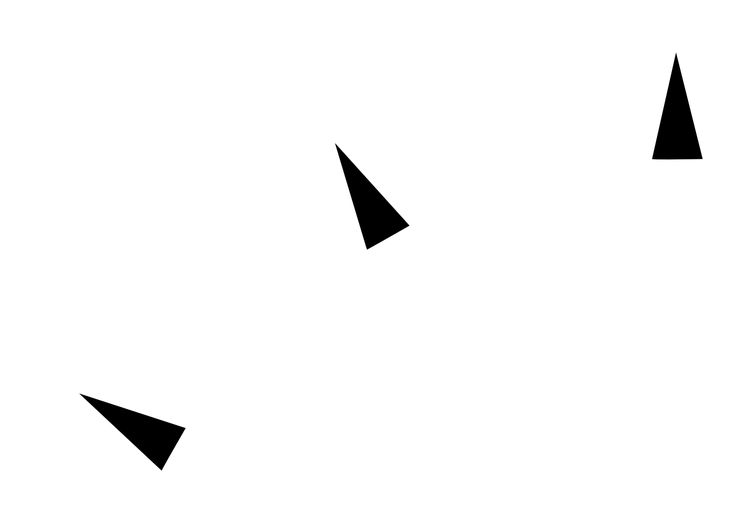
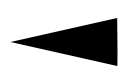
Solaris Handbuch für fortgeschrittene Benutzer



Sun Microsystems GmbH
Bretonischer Ring 3
85630 Grasbrunn
Tel.: (0 89) 4 60 08-0
Fax: (0 89) 4 60 08-2 22





Bestellnummer: 801-7482-10
Version 50, *August 1994*

Inhalt

Einführung

Angesprochener Leserkreis

Dieses Handbuch richtet sich an Benutzer der Solaris™ Systemsoftware. Solaris besteht aus dem Betriebssystem SunOS™ und der grafischen Benutzeroberfläche OpenWindows™.

Voraussetzungen

Die Systemsoftware sollte bereits installiert und betriebsbereit sein. Ist dies noch nicht geschehen, finden Sie die nötigen Informationen in dem Handbuch *Sparc: Solaris Software-Installation* bzw. im Handbuch *x86: Solaris Software-Installation* sowie in dem mit Ihrem System gelieferten Installationshandbuch.

Weitere Handbücher

Über die Anwendung „AnswerBook" können Sie auf Informationsmaterial zu Solaris zugreifen, das in folgenden Buchreihen zusammengefaßt ist:

- Sun Administrator's Set
  Enthält ausführliche Informationen zur Installation und Verwaltung verschiedener Systemkonfigurationen, einschließlich größerer Sun-Workstation-Netzwerke .

- Sun Developer's Set
  Enthält ausführliche Informationen zur Entwicklung, Fehlerbeseitigung und Pflege von Anwendungen, die unter Solaris laufen.

- Sun Reference Manual Set
  Enthält Beschreibungen der SunOS Befehle, die als `man` pages vorliegen und bei Bedarf auch als Online-Dokumentation installiert werden können.

- Sun User's Set
  Enthält Informationen zu folgenden Themenbereichen: SunOS, Verwendung der SunOS Befehle, OpenWindows, Anpassung des Arbeitsbereichs, Fehlerbeseitigung, Erstellung von Shell-Skripts, Versenden von elektronischer Post und Arbeiten im Netzwerk.

Weitere Informationen zu der Solaris AnswerBook-Software finden Sie im *Solaris Roadmap*.

Richtlinien zur Schreibweise

In diesem Handbuch gelten folgende Richtlinien zur Schreibweise:

- Wichtige Begriffe, Befehlsargumente, Variablen und Verweise auf weitere Dokumentation sind *kursiv* gedruckt. Zum Beispiel:

  Erläuterungen zur Eingabe des Arguments *dateiname* finden Sie in den *man Pages(1): User Commands* (Online-Dokumentation zu SunOS-Befehlen).

- **`In dieser Schriftart`** erscheinen Informationen, die Sie während einer Beispielsitzung eingeben. Zum Beispiel:

```
$ date
```

- `Diese Schriftart` wird für Listings sowie für Befehls- und Systemnamen verwendet. Zum Beispiel:

  Um sich als Superuser anzumelden, weisen Sie sich bei der Anmeldung als `root` aus oder geben den Befehl `su` in die Kommandozeile ein.

In dieser Schriftart erscheinen außerdem Meldungen, die das System generiert und auf den Bildschirm ausgibt. Zum Beispiel:

```
$ who
bozo     console     Sep 11 15:36
```

- Die Namen der Maustasten erscheinen in Großbuchstaben und verweisen auf die Funktion, nicht auf die Position der Tasten (also „Drücken Sie AUSWAHL“ und nicht etwa „Drücken Sie LINKS“). Ursache dafür ist, daß die Belegung der Maustasten geändert werden kann und die linke Maustaste daher nicht immer mit der Funktion AUSWAHL belegt sein muß.

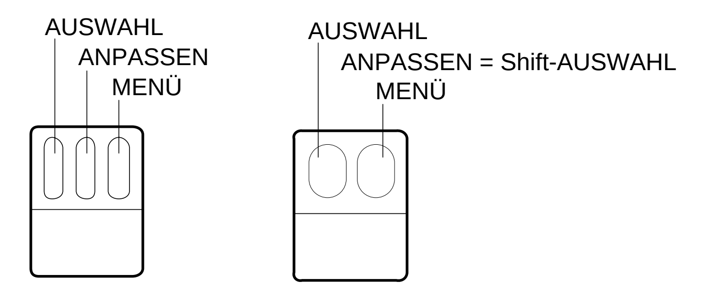


Abbildung P-1 Standardbelegung der Maustasten

Bei Mäusen mit drei Tasten sind die Maustasten standardmäßig wie folgt belegt:

- AUSWAHL = linke Maustaste
- ANPASSEN = mittlere Maustaste
- MENÜ = rechte Maustaste

Bei Mäusen mit zwei Tasten sind die Maustasten standardmäßig wie folgt belegt:

- AUSWAHL = linke Maustaste
- MENÜ = rechte Maustaste

Um die Funktion ANPASSEN auf einer Maus mit zwei Tasten auszuführen, drücken Sie die Taste „Shift“ auf der Tastatur, halten sie gedrückt und drücken gleichzeitig die Maustaste AUSWAHL.

- Tastennamen erscheinen in Anführungszeichen. (Beispiel: Drücken Sie „Return“.) Auf manchen Tastaturen hat die Taste „Return“ die Beschriftung „Enter“.

Hinweis – Solaris unterstützt sowohl SPARC-Systeme als auch x86-Systeme. Die Tastaturen dieser Systeme sind unterschiedlich. Daher sind die Tastaturbefehle für einige OpenWindows-Menübefehle je nach Tastatur ebenfalls unterschiedlich.

Für die meisten Tastaturbefehle gibt es jedoch unter Solaris eine zusätzliche Tastenkombination mit der Meta-Taste, die auf beiden Plattformen benutzt werden kann. Kapitel 1, „Anmelden bei SunOS und Starten von OpenWindows" enthält eine Liste der OpenWindows-Befehle und der verschiedenen äquivalenten Tastaturbefehle.

- OpenWindows-Menübefehle, die unter Verwendung von Tastenkombinationen ausgeführt werden, sind in diesem Handbuch wie folgt dargestellt: Ein Bindestrich (-) zwischen zwei Tastenbezeichnungen zeigt an, daß die beiden Tasten gleichzeitig gedrückt werden müssen.

  Beispiel: „Strg"-„Alt" bedeutet, daß Sie die Tasten „Strg" und „Alt" gleichzeitig drücken und niederhalten müssen; „Strg"-„Alt"-„c" bedeutet, daß Sie die Tasten „Strg" und „Alt" gleichzeitig drücken und niederhalten, und zugleich das Zeichen „c" eingeben.

- Tastenkombinationen mit der Taste „Meta" geben Sie genau wie OpenWindows-Menübefehle ein, indem Sie die Taste „Meta" und eine oder mehrere Tasten auf der Tastatur drücken. Auf SPARC-Tastaturen ist die Taste „Meta" durch eine Raute (<>) gekennzeichnet. Auf x86-Tastaturen erzeugen Sie die Funktion dieser Taste durch gleichzeitiges Drücken der Tasten „Strg" und „Alt" („Strg"-„Alt"). Ausführliche Hinweise zu Tastaturabkürzungen und Funktionstasten finden Sie in Kapitel 1, „Anmelden bei SunOS und Starten von OpenWindows".

- Wenn im Text auf ein gesamtes Menü oder auf die Standardoption eingegangen wird, ist die *Standardoption* auf der Abbildung *umrandet*. OpenWindows ist standardmäßig so eingestellt, daß das gesamte Menü angezeigt wird, wenn Sie mit der linken Maustaste auf eine Menütaste klicken. Sie können diese Standardeinstellung auch ändern, so daß die Standardoption aus einem Menü ausgewählt und das Menü nicht angezeigt wird, wenn Sie mit der linken Maustaste auf die Menütaste klicken. In dem vorliegenden Handbuch wird davon ausgegangen, daß Sie die letztere Einstellung gewählt haben.

- Bei der Auswahl von Menüoptionen wird folgende Schreibweise verwendet: Beispiel: Wählen Sie Arbeitsbereich ➤ Programme.

Plattformspezifische Vereinbarungen

Die folgenden Symbole am Anfang eines Abschnitts bzw. vor dem Titel eines Arbeitsschritts weisen darauf hin, daß der Abschnitt bzw. Arbeitsschritt sich nur auf die angegebene Plattform bezieht.

SPARC Bezeichnet einen SPARC-plattformspezifischen Abschnitt bzw. Arbeitsschritt

x86 Bezeichnet einen x86-plattformspezifischen Abschnitt bzw. Arbeitsschritt

Plattformspezifische Hinweise sind wie folgt gekennzeichnet:

SPARC – Dieser Hinweis gilt nur für SPARC-Plattformen.

---

x86 – Dieser Hinweis gilt nur für x86-Plattformen.

Plattformspezifische Absätze mit Blickfangpunkten (Bullets) sind wie folgt gekennzeichnet:

- **SPARC:** Dieser Absatz gilt nur für SPARC-Plattformen.
- **x86:** Dieser Absatz gilt nur für x86-Plattformen.

Falls Code-Beispiele, Tabellen oder Abbildungen nur für eine bestimmte Plattform gelten, ist der Name der Plattform vor dem Titel angegeben. Beispiel:

Tabelle 1-1 SPARC: Tabellentitel (spezifisch für SPARC-Plattform)

Tabelle 1-2 x86: Tabellentitel (spezifisch für x86-Plattform)

Anmelden bei SunOS und Starten von OpenWindows *1* 

Voraussetzung ist, daß OpenWindows Version bereits auf Ihrer Festplatte oder auf einem erreichbaren Server installiert ist. Wenn Sie sich darüber nicht sicher sind, sollten Sie Ihren Systemverwalter fragen oder im Handbuch *Sparc: Solaris Software-Installation* oder in *x86: Solaris Software-Installation* nachschlagen.

Dieses Kapitel beschreibt, wie Sie sich beim System anmelden, Befehle in eine Shell eingeben, die OpenWindows-Umgebung von einem Server in das lokale Dateisystem einhängen, OpenWindows starten und verlassen und sich beim System abmelden. Außerdem erfahren Sie, welche Befehlsoptionen Sie beim Start von OpenWindows auf speziell konfigurierten Systemen, wie z. B. einer Workstation mit zwei Bildschirmen, eingeben müssen.

Anmelden

Der Zeitraum von der Anmeldung beim System bis zur Abmeldung wird als *Sitzung* bezeichnet. Da SunOS ein Mehrbenutzer-Betriebssystem ist, müssen Sie sich bei der Anmeldung eindeutig ausweisen, damit zwischen den einzelnen Benutzern unterschieden werden kann. Dazu geben Sie Ihren *Benutzernamen* (auch als *Login-Name* oder *Zugangsberechtigung* bezeichnet) ein. Weil sich jedoch unter diesem Namen auch andere Benutzer anmelden könnten, wird zusätzlich ein *Paßwort* abgefragt, das nur Ihnen bekannt ist. Wenn Sie noch keinen Benutzernamen und kein Paßwort erhalten haben, bitten Sie den Systemverwalter, eine Zugangsberechtigung für Sie einzurichten. Danach können Sie sich bei SunOS anmelden.

Bevor Sie sich beim System anmelden, sollte der Bildschirm etwa so aussehen:

```
login:
```

Geben Sie den Benutzernamen ein, den Sie von Ihrem Systemverwalter erhalten haben, und drücken Sie „Return". Zum Beispiel:

```
login: nina
```

Nach Abschluß der Eingabe mit „Return" wird das Paßwort abgefragt:

```
login: nina
Password:
```

Geben Sie das Paßwort ein, und drücken Sie „Return". (Wenn zu Ihrem Benutzernamen kein Paßwort gehört, wird es auch nicht abgefragt.)
Aus Sicherheitsgründen erscheint das Paßwort bei der Eingabe nicht auf dem Bildschirm.

Wenn Ihnen kein Paßwort zugewiesen wurde und Sie eines haben möchten, oder wenn Sie das vorhandene Paßwort ändern wollen, lesen Sie bitte den Abschnitt „Allgemeine Hinweise zu Paßwörtern" in Kapitel 5, „Paßwörter und Prozesse".

Login-Shell

In den folgenden Kapiteln werden Sie verschiedene SunOS Befehle kennenlernen. Bei der Eingabe eines Systembefehls übergeben Sie Informationen an einen Befehlsinterpreter, der als *Shell* bezeichnet wird. Das Shell-Programm liest diese Informationen und löst die entsprechende Aktion auf Systemebene aus.

SunOS verwendet standardmäßig die Bourne-Shell. Außer dieser Shell sind jedoch noch zwei weitere verfügbar: die C-Shell und die Korn-Shell. Jedes dieser Shell-Programme weist bestimmte Unterschiede auf.

Hinweis – Wenn Sie Informationen zu einem SunOS-Befehl oder einer der verfügbaren Shells benötigen, können Sie die dazugehörige `man` page (Online-Dokumentation zu Betriebssystem-Befehlen) aufrufen. Hinweise dazu finden Sie im Abschnitt „man pages (Online-Dokumentation zu SunOS-Befehlen) anzeigen“ in Kapitel 2, „SunOS-Befehle“.

Wenn Sie sich beim System anmelden (oder ein neues Kommando- oder UNIX-Shell-Fenster öffnen), wird automatisch ein Shell-Programm gestartet. Dieses Programm heißt *Login-Shell*. Handelt es sich dabei nicht um die standardmäßig verwendete Bourne-Shell, hat der Systemverwalter bei der Konfiguration eine andere Shell (C oder Korn) angegeben.

Wie bereits erwähnt, verhalten sich diese Shells unterschiedlich. Manche Befehle oder Prozeduren, die in einer Shell zur Verfügung stehen, werden von einer anderen nicht unterstützt. Die Beispiele in diesem Handbuch beziehen sich im allgemeinen auf die Bourne-Shell. Werden Befehle oder Prozeduren verwendet, die in der Bourne-Shell nicht verfügbar sind, weisen wir ausdrücklich darauf hin.

Abmelden

Wenn Sie Ihre Arbeitssitzung beenden und das Betriebssystem verlassen wollen, geben Sie folgenden Befehl ein:

```
$ exit
```

Daraufhin erscheint wieder die Anmelde-Eingabeaufforderung:

```
$ exit
login:
```

Sobald diese Eingabeaufforderung angezeigt wird, sind Sie abgemeldet. Um mit dem System weiterarbeiten zu können, müssen Sie sich neu anmelden.

 *1*

Hinweis – Durch Ausschalten Ihrer Workstation oder Ihres Terminals melden Sie sich *nicht* automatisch ab, d. h. nach dem Wiedereinschalten kann es sein, daß Sie weiterhin als Benutzer angemeldet sind. Unter SunOS müssen Sie sich explizit abmelden, um die Verbindung zum System abzubauen.

Tastaturbefehle

In vielen Fällen können Sie durch Eingabe von *Tastaturbefehlen* die Ausführung bestimmter Operationen beschleunigen. Durch Eingabe bestimmter Tastenkombinationen und Verwendung vorkonfigurierter Tasten auf der Tastatur erübrigt sich die Verwendung der Maus für das Aufrufen von Menüs und das Anklicken von Optionen.

Die folgende Tabelle enthält eine Liste der wichtigsten Operationen und der entsprechenden Tastaturbefehle auf SPARC- sowie x86-Systemen.

Hinweis – Die Taste „Meta" ist auf SPARC-Tastaturen durch eine Raute (<>) gekennzeichnet. Auf x86-Tastaturen führen Sie die Funktion der Taste „Meta" aus, indem Sie gleichzeitig die Tasten „Strg" und „Alt" drücken („Strg"-„Alt").

Um einen Tastaturbefehl auszuführen, drücken Sie die erste Taste und halten sie gedrückt („Meta" bzw. „Strg"-„Alt"), und geben dann das zweite Zeichen ein. Um beispielsweise markierten Text auszuschneiden, drücken und halten Sie „Meta" gedrückt und drücken dann die Taste *x* auf Ihrer SPARC-Tastatur. Auf einem x86-System drücken und halten Sie „Strg"-„Alt" gedrückt und drücken gleichzeitig *x*.

Table 1-1 Tastaturbefehle

Operation	Tastaturbefehl	Aktion
Ausschneiden	Meta - x	Schneidet die Auswahl aus und legt sie in der Zwischenablage ab
Drucken	Meta - p	Schickt die Datei zum Drucker (z. B. die Datei, deren Symbol Sie im Dateimanager markiert haben)
Eigenschaften	Meta - i	Zeigt das Fenster „Eigenschaften“ für die Anwendung an der Zeigerposition an
Einfügen	Meta - v	Kopiert den Inhalt aus der Zwischenablage und fügt ihn an der Position der Einfügemarke ein
Erneut	Shift-Meta - p	Macht Zurücknehmen rückgängig
Hilfe	Help or F1	Zeigt ein Hilfefenster mit kontextbezogener Hilfe für das Objekt an der Zeigerposition an
Kopieren	Meta - c	Kopiert die Auswahl in die Zwischenablage
Neu	Meta - n	Öffnet eine neue Datei
Öffnen (Datei)	Meta - o	Öffnet eine Datei (z. B. die Datei, deren Symbol Sie im Dateimanager markiert haben)
Öffnen (Fenster)	Meta - w	Öffnet ein Symbol bzw. schließt ein Fenster zu einem Symbol
Sichern	Meta - s	Sichert die aktuelle Datei
Stop	Stop or Esc	Stoppt die aktuelle Operation
Suchen	Meta - f	Sucht den nächsten rechts von der Einfügemarke stehenden Suchbegriff
Wiederholen	Meta - a	Wiederholt die letzte Operation
Zurücknehmen	Meta - u	Macht die letzte Operation rückgängig

 *1*

Vorbereitungen für den Start von OpenWindows

Bevor Sie die OpenWindows-Software starten, sollten Sie prüfen, ob die folgenden Hinweise auf Sie zutreffen. Wenn nicht, können Sie mit dem Abschnitt „OpenWindows starten“ in diesem Kapitel fortfahren.

- Wenn Sie für die Konfiguration des OpenWindows-Servers verantwortlich sind, schlagen Sie bitte im Handbuch *Sparc: Solaris Software-Installation* oder *x86: Solaris Software-Installation* nach.
- Wenn Sie bis jetzt mit OpenWindows Version 2 gearbeitet haben, lesen Sie bitte Anhang A, „Aktualisierung auf OpenWindows Version 3.3 oder höhere Versionen“,.
- Wenn Sie bis jetzt mit der SunView-Benutzeroberfläche gearbeitet haben, lesen Sie bitte Anhang A, „Aktualisierung auf OpenWindows Version 3.3 oder höhere Versionen“,.
- Wenn Sie derzeit mit einer OpenWindows-Version arbeiten, die älter als Version 3.3 ist, lesen Sie bitte die Abschnitte „Die Umgebungsvariable OPENWINHOME“ und „Die Datei .xinitrc“ in diesem Kapitel..

---

x86 – Falls Sie nach der Installation von Solaris auf Ihrem System den Gerätetyp der Maus, der Tastatur oder der Video-Adapterkarte geändert haben, müssen Sie das Programm devconfig erneut ausführen, um die Systemkonfiguration zu aktualisieren. Siehe hierzu auch die `man` page devconfig(1M).

Die Umgebungsvariable `OPENWINHOME`

Wenn Sie noch mit einer OpenWindows-Version arbeiten, die älter als Version`3.3 ist, haben Sie wahrscheinlich die Umgebungsvariable `OPENWINHOME` in einer Konfigurationsdatei gesetzt. Dies wird jedoch, ebenso wie das Setzen der Variablen mit dem Befehl `setenv`, nicht mehr empfohlen.

Bei Eingabe des Befehls `openwin` wird die Umgebungsvariable `OPENWINHOME` jetzt automatisch auf den Pfad `/usr/openwin` gesetzt, so daß weitere Schritte nicht erforderlich sind.

Wenn Sie die Variable OPENWINHOME in der Datei .profile oder .cshrc in Ihrem Home-Verzeichnis gesetzt haben, löschen Sie diese Zeile oder markieren Sie sie als Kommentar, *bevor* Sie OpenWindows Version`3.3 oder eine spätere Version starten.

Dazu gehen Sie folgendermaßen vor:

1. **Öffnen Sie die Datei** .profile **oder** .cshrc **mit einem Texteditor (z. B.** vi**).**

2. **Geben Sie vor der Variablen, wie unten gezeigt, ein Nummernzeichen (#) ein, oder löschen Sie die gesamte Zeile aus der Datei.**
   Falls Sie die Datei .profile bearbeiten, befolgen Sie Schritt a; falls sie die Datei .cshrc bearbeiten, befolgen Sie Schritt b.

   a. **In der Datei** .profile **sollte die Zeile danach so aussehen:**

```
#OPENWINHOME=/usr/openwin
```

   b. **In der Datei** .cshrc **sollte die Zeile danach so aussehen:**

```
#setenv OPENWINHOME /usr/openwin
```

3. **Speichern Sie die Datei, und verlassen Sie den Texteditor.**

4. **Setzen Sie die Umgebungsvariable** OPENWINHOME **zurück.**
   Statt den unten angegebenen Befehl einzugeben, können Sie sich auch beim System ab- und dann wiederanmelden.

   a. **In der Bourne- oder Korn-Shell geben Sie folgenden Befehl ein:**

```
$ unset OPENWINHOME
```

   b. **In der C-Shell geben Sie folgenden Befehl ein:**

```
example% unsetenv OPENWINHOME
```

Danach können Sie die OpenWindows-Software starten. Hinweise dazu finden Sie im Abschnitt „OpenWindows starten“.

Die Datei `.xinitrc`

Wenn Sie bisher mit einer älteren OpenWindows-Version als 3.3 gearbeitet haben, sollten Sie den Inhalt der Konfigurationsdatei `.xinitrc` in Ihrem Home-Verzeichnis prüfen. Diese Datei enthält unter Umständen benutzer-spezifische Optionen.

Um den Status der Datei`.xinitrc` zu prüfen, geben Sie folgende Befehle ein:

```
$ cd
$ ls -a .xinitrc
```

Danach gehen Sie, je nach Befehlsausgabe, folgendermaßen vor:

- Ist eine Datei namens.xinitrc nicht vorhanden (der Befehl ls -a gibt kein Listing aus), brauchen Sie nichts zu tun, da OpenWindows in diesem Fall die Standard-Konfigurationsdatei verwendet.
- Existiert die Datei .xinitrc (der Befehl ls -a gibt ein Datei-Listing aus), die jedoch nicht bearbeitet wurde oder keine Änderungen enthält, die übernommen werden sollen, führen Sie Schritt 1 im Abschnitt „.xinitrc löschen oder umbenennen" aus.
- Existiert die Datei.xinitrc und enthält sie Änderungen, die erhalten bleiben sollen, führen Sie Schritt 2 im Abschnitt „.xinitrc löschen oder umbenennen" aus.

`.xinitrc` *löschen oder umbenennen*

1. **Um die Datei**`.xinitrc` **aus Ihrem Home-Verzeichnis zu löschen, geben Sie folgenden Befehl ein:**

```
$ rm .xinitrc
```

2. **Um die Änderungen in der Datei** `.xinitrc` **zu übernehmen, gehen Sie wie folgt vor:**

a. Benennen Sie `.xinitrc` **in** `.xinitrc.save` **um:**

```
$ mv .xinitrc .xinitrc.save
```

b. Kopieren Sie `$OPENWINHOME/lib/Xinitrc` **in** `.xinitrc` **in Ihrem Home-Verzeichnis:**

```
$ cp $OPENWINHOME/lib/Xinitrc $HOME/.xinitrc
```

c. Kopieren Sie die gewünschten Zeilen aus `.xinitrc.save` **in** `.xinitrc`**.**



Vorsicht – Die neue `.xinitrc`-Datei darf nur einen Aufruf für `olwm` und keinen Aufruf für `svenv` enthalten. Die Zeile `$OPENWINHOME/lib/openwin-sys` darf nicht gelöscht werden.

OpenWindows-Umgebung einrichten

Um die OpenWindows-Benutzerumgebung einzurichten, gehen Sie wie folgt vor:

1. Hängen Sie die OpenWindows-Software mit NFS von dem bereitstellenden Server in das lokale Dateisystem ein.

   Wenn Sie Informationen zu diesem Schritt benötigen, schlagen Sie im *OpenWindows Reference Manual* nach oder wenden sich an Ihren Systemverwalter.

2. Starten Sie OpenWindows mit dem Befehl `openwin` und, falls erforderlich, speziellen Befehlsoptionen.

OpenWindows starten

Sobald Sie die OpenWindows-Software von einem Server eingehängt und das Installations-Skript aufgerufen haben, können Sie OpenWindows starten.

Dazu geben Sie in einer Shell `/usr/openwin/bin/openwin` ein und drükken „Return“.

```
$ /usr/openwin/bin/openwin
```

Daraufhin wird der Eingangsbildschirm von OpenWindows Version 3.3 angezeigt und der Arbeitsbereich konfiguriert.

OpenWindows direkt starten

Sobald Sie OpenWindows einmal erfolgreich gestartet haben, können Sie Ihr System so konfigurieren, daß Sie beim Start nicht jedesmal den vollständigen Pfad zu den OpenWindows-Dateien eingeben müssen.

Wenn Sie mit einer Bourne- oder Korn-Shell arbeiten, nehmen Sie dazu eine Shell-Funktion in die Datei `.profile` auf. Bei Verwendung der C-Shell setzen Sie einen *Alias* in die Datei `.cshrc`. Beide Dateien befinden sich in Ihrem Home-Verzeichnis.

Danach können Sie OpenWindows einfach mit folgender Eingabe starten:

```
$ openwin
```

In den beiden folgenden Abschnitten, „Bearbeitung der Datei .profile“ und „Bearbeiten der Datei .cshrc“, erfahren Sie, wie Sie den OpenWindows-Verweis in die für Ihre Shell geeignete Konfigurationsdatei aufnehmen.

Bearbeitung der Datei `.profile`

So nehmen Sie den OpenWindows-Verweis in die Datei `.profile` auf:

1. **Öffnen Sie die Datei `.profile` mit einem Texteditor (z. B. `vi`).**

2. **Fügen Sie die unten gezeigte Shell-Funktion in die Datei ein:**

```
openwin () {
        /usr/openwin/bin/openwin
}
```

3. **Sichern Sie die Datei, und verlassen Sie den Texteditor.**
4. **Melden Sie sich ab und wieder an, um die neue Funktion zu aktivieren, oder geben Sie folgenden Befehl ein:**

```
$ . .profile
```

Nun können Sie OpenWindows durch Eingabe von `openwin` starten.

Bearbeiten der Datei `.cshrc`

So nehmen Sie den OpenWindows-Verweis in die Datei `.cshrc` auf:

1. **Öffnen Sie die Datei** `.cshrc` **mit einem Texteditor (z. B.** `vi`**).**
2. **Fügen Sie folgende Zeile in die Datei ein:**

```
alias openwin /usr/openwin/bin/openwin
```

3. **Sichern Sie die Datei, und verlassen Sie den Texteditor.**
4. **Melden Sie sich ab und wieder an, oder geben Sie folgenden Befehl ein:**

```
example% source .cshrc
```

Nun können Sie OpenWindows durch Eingabe von `openwin` starten.

Wenn sich OpenWindows nicht starten läßt

Beim Start von OpenWindows greift das System auf das Verzeichnis `/usr/openwin` zu. In diesem Verzeichnis wird OpenWindows standardmäßig installiert. Viele Anwendungen, z. B. der Terminkalender, können nicht geladen werden, wenn sich die OpenWindows-Software nicht im Verzeichnis `/usr/openwin` befindet.

Läßt sich OpenWindows durch Eingabe von `/usr/openwin/bin/openwin` nicht starten, ist die OpenWindows-Software entweder noch gar nicht oder in einem anderen Verzeichnis als `/usr/openwin` installiert. Schlagen Sie in diesem Fall im *OpenWindows Reference Manual* nach oder wenden Sie sich an Ihren Systemverwalter.

OpenWindows beenden

Wenn Sie sich in der OpenWindows-Umgebung befinden, müssen Sie diese zunächst verlassen, bevor Sie sich beim System abmelden können.

Wenn Sie innerhalb der OpenWindows-Umgebung ein Shell-Fenster öffnen und den Befehl `logout` eingeben, erscheint folgende Meldung:

```
Dies ist nicht die Login-Shell.
```

Um OpenWindows zu beenden, gehen Sie folgendermaßen vor:

1. **Bewegen Sie den *Mauszeiger* auf den *Arbeitsbereich*, d. h. auf einen Teil des Bildschirms, an dem sich keine Fenster oder Symbole befinden.**

2. **Drücken Sie die Maustaste MENÜ.**
   Das Menü „Arbeitsbereich“ wird geöffnet.

3. **Ziehen Sie den Zeiger auf die Menüoption „Verlassen“.**

4. **Lassen Sie die Maustaste los.**
   Sie werden in einem Popup-Fenster zur Bestätigung Ihrer Auswahl aufgefordert.

5. **Bewegen Sie den Zeiger auf die Taste „Verlassen“, und klicken Sie die Maustaste AUSWAHL.**
   Kurz darauf werden alle Fenster geschlossen, und in der unteren linken Ecke des Bildschirms erscheint die Eingabeaufforderung des Systems.

Spezielle Startoptionen

Normalerweise genügt es, wenn Sie OpenWindows mit der im Abschnitt „OpenWindows starten“ beschriebenen Methode aufrufen. In bestimmten Fällen müssen Sie jedoch spezielle Optionen angeben, um die Konfiguration der OpenWindows-Umgebung beim Start zu ändern.

In diesem Abschnitt werden einige dieser Sonderfälle behandelt:

- OpenWindows soll mit eingeschränktem Zugriffsschutz gestartet werden.
- OpenWindows soll auf einem System mit Graustufenbildschirm laufen.
- OpenWindows soll auf einem System mit mehreren Bildschirmen gestartet werden.

Die Befehlsoptionen werden nach dem Befehl `openwin` eingegeben:

```
$ openwin [ Optionen ]
```

Auf dieser und den folgenden Seiten werden einige dieser Optionen ausführlicher beschrieben.

Eingeschränkter Zugriffsschutz

Wenn Sie in einer offenen Netzwerkumgebung arbeiten, in der Datenschutz keine große Rolle spielt, können Sie mit der Option `-noauth` anderen Benutzern gestatten, Anwendungen auf Ihrem System zu starten.

Mit folgendem Befehl deaktivieren Sie den Zugriffsschutz und geben damit anderen Benutzern die Möglichkeit, auf Ihren Window Server zuzugreifen:

```
$ openwin -noauth
```

Systeme mit Graustufen-Bildschirm

Wenn Sie mit einem Graustufen-Bildschirm (8 Bit oder mehr Pixeltiefe) arbeiten, können Sie OpenWindows mit der Option `grayvis` starten. Dies ist zwar nicht unbedingt notwendig, verbessert jedoch mitunter die Bildqualität:

```
$ cd
$ openwin -dev /dev/fb grayvis
```

SPARC Systeme mit mehreren Bildschirmen

SPARC – Beachten Sie, daß sich die folgenden Abschnitte zum Thema „Systeme mit mehreren Bildschirmen“ nur für Systeme mit SPARC-Prozessor gelten. Wenn Sie bisher noch nie ein System konfiguriert haben, sollten Sie Ihren Systemverwalter bitten, Ihnen bei der Ausführung der folgenden Schritte zu helfen.

Damit OpenWindows auf einem System mit mehreren Bildschirmen ordnungsgemäß läuft, müssen Sie beim Start die Namen dieser Geräte und die erforderlichen Anzeigeoptionen angeben.

Um die OpenWindows-Software auf einem System mit zwei Bildschirmen zu starten, geben Sie den Befehl `openwin` mit folgenden Optionen ein:

```
$ openwin [ [ -dev gerät ] [ geräte_optionen ] ]
```

Die eckigen Klammern geben an, daß die Kombination [`-dev` *gerät] [geräte_optionen*] auch mehrmals (d. h. einmal pro Gerät) eingegeben werden kann.

[-dev gerät]

Mit der Option `-dev` teilen Sie dem Server den Gerätenamen des Bildschirms mit. Wenn Sie diese Option nicht angeben, verwendet der Server den Standard-Gerätenamen `/dev/fb`. Soll OpenWindows auf mehreren Bildschirmen laufen, geben Sie die Option `-dev` entsprechend oft ein.

[geräte_optionen]

Die *geräte_optionen* sind zusätzliche Argumente, mit denen Sie die Eigenschaften des mit der Option `-dev` spezifizierten Geräts näher beschreiben.

Beispiele für Geräte-Optionen

Dieser Abschnitt beschreibt, welche Optionen Sie eingeben müssen, wenn OpenWindows auf einem System mit zwei nebeneinander oder übereinander angeordneten Bildschirmen laufen soll.

Hinweis – Die Reihenfolge der Geräte in der Befehlszeile ist von entscheidender Bedeutung. Wenn Sie die Reihenfolge nicht mit den Argumenten `left` und `right` angeben, muß das links oder oben stehende Gerät stehende Gerät als erstes genannt werden.

[left] [right]

Mit dem folgenden Befehl starten Sie OpenWindows auf einem System mit zwei nebeneinanderstehenden Bildschirmen, von denen der rechte ein Schwarzweiß-Bildschirm ist. Nach dem Start können Sie den Mauszeiger zwischen den beiden Bildschirmen hin- und herbewegen.

```
$ openwin -dev /dev/fb left -dev /dev/fbs/bwtwo0 right
```

Da das in der Befehlszeile zuerst genannte Gerät standardmäßig immer links steht, können Sie die Argumente `left` und `right` in diesem Fall auch weglassen:

```
$ openwin -dev /dev/fb -dev /dev/fbs/bwtwo0
```

Mit dem folgenden Befehl starten Sie OpenWindows auf einem System mit zwei nebeneinanderstehenden Bildschirmen, von denen der linke ein Schwarzweiß-Bildschirm ist:

```
$ openwin -dev /dev/fb right -dev /dev/fbs/bwtwo0 left
```

[top] [bottom]

Mit dem folgenden Befehle starten Sie OpenWindows auf einem System mit zwei aufeinanderstehenden Bildschirmen, von denen der obere ein CG6-Bildschirm und der untere ein Schwarzweiß-Bildschirm ist. Nach dem Start können Sie den Mauszeiger zwischen den beiden Bildschirm hin- und herbewegen.

```
$ openwin -dev /dev/fbs/cgsix0 top -dev /dev/fbs/bwtwo0 bottom
```

Der nächste Befehl ist *nicht* mit dem vorhergehenden identisch, da das zuerst genannte Gerät standardmäßig immer *links* von dem zweiten Gerät steht:

```
$ openwin -dev /dev/fbs/cgsix0 -dev /dev/fbs/bwtwo0
```

Mit dem folgenden Befehl starten Sie OpenWindows auf einem System mit zwei aufeinanderstehenden Bildschirmen, von denen der untere ein CG6-Bildschirm und der obere ein Schwarzweiß-Bildschirm ist:

```
$ openwin -dev /dev/fbs/cgsix0 bottom -dev /dev/fbs/bwtwo0 top
```

Weitere Hinweise

Folgende Hinweise sind zu beachten, wenn Sie OpenWindows auf mehreren Bildschirmen ablaufen lassen:

- Normalerweise werden alle Bildschirme durch olwm verwaltet.
- Fenster können nicht zwischen Bildschirmen verschoben werden.

SunOS-Befehle 2



Dieses Kapitel führt Sie in die Verwendung der SunOS Benutzerbefehle ein. Sie erfahren unter anderem, wie Sie Befehle eingeben, Eingabefehler korrigieren, lange Befehle oder mehrere Befehle auf einmal eingeben und welche Arten von Befehlsoptionen zur Verfügung stehen.

SunOS-Befehle werden in einer Shell eingegeben. Um unter OpenWindows ein Shell-Fenster zu öffnen, zeigen Sie das Menü „Arbeitsbereich" an und wählen im Untermenü „Programme" die Option „Kommando-Shell" oder „UNIX-Shell".

Eingabeaufforderungen

Nach der Anmeldung beim System ist der Bildschirm oder das Shell-Fenster bis auf eine Eingabeaufforderung leer. Wie diese Eingabeaufforderung aussieht, hängt davon ab, mit welcher Shell Sie arbeiten und wie diese vom Systemverwalter konfiguriert wurde. In den Beispielen dieses Handbuchs wird die Standard-Eingabeaufforderung der SunOS Systemsoftware verwendet: ein Dollarzeichen ($).

Wenn Sie die Eingabeaufforderung zu einem späteren Zeitpunkt ändern möchten, lesen Sie bitte den Abschnitt „Eingabeaufforderung ändern" in Kapitel 10, „Anpassen der Arbeitsumgebung".

 2

Befehle eingeben

Die Eingabeaufforderung gibt an, daß das System bereit ist, Befehle entgegenzunehmen und zu verarbeiten. Geben Sie probehalber den Befehl `date` ein, und schließen Sie die Eingabe mit „Return“ ab:

```
$ date
Montag, 3. Februar 1992, 10:12:51 Uhr MET
$
```

Dieser Befehl zeigt das aktuelle Datum und die aktuelle Uhrzeit an. Geben Sie den gleichen Befehl jetzt mit großem Anfangsbuchstaben ein:

```
$ Date
Date: Befehl nicht gefunden.
$
```

Wie Sie sehen, ist ein Großbuchstabe für das System nicht dasselbe wie ein Kleinbuchstabe. SunOS Befehle werden fast alle in Kleinbuchstaben eingegeben.

Eingabefehler korrigieren

Nehmen wir an, Sie haben `Date` zwar schon eingegeben, Ihren Fehler jedoch bemerkt, bevor Sie „Return“ gedrückt haben. Da der Befehl erst nach dem Drücken von „Return“ an das System gesendet wird, können Sie den Fehler noch korrigieren. Dafür stehen zwei Methoden zur Auswahl:

- Sie drücken die Rücktaste (Back Space) oder die Taste „Del“ so oft, bis das falsch eingegebene Zeichen gelöscht ist.
- Sie drücken „Ctrl-U“, um die gesamte Zeile zu löschen.

Probieren Sie beide Methoden aus. Die Funktion der Rücktaste und der Taste „Del“ ist nicht auf allen Systemen gleich. Die Tastenkombination „Ctrl-U“ funktioniert auf den meisten Systemen.

Mehrere Befehle und lange Befehle eingeben

Wenn Sie mehrere Befehle gleichzeitig an das System übergeben wollen, verwenden Sie das Semikolon als Trennzeichen. Wenn Sie die Befehle `date` und `logname` wie in folgendem Beispiel gezeigt eingeben, brauchen Sie zum Abschluß nur einmal „Return“ zu drücken:

```
$ date; logname
Montag, 3. Februar 1992, 10:12:51 Uhr MET
nina
$
```

Da die Befehle nacheinander abgearbeitet werden, erscheinen zunächst das Datum und die Uhrzeit (die Ausgabe des Befehls `date`) und danach der Login-Name des aktuell angemeldeten Benutzers (Ausgabe des Befehls `logname`).

Wenn Sie einen sehr langen Befehl eingeben, können Sie durch Eingabe eines Gegenschrägstrichs (\) eine neue Zeile beginnen. Zum Beispiel:

```
$ date; \
logname
Montag, 3. Februar 1992, 10:12:51 Uhr MET
nina
$
```

Die Befehle `date` und `logname` sind zwar nicht gerade lang, aber sie sollen hier auch nur demonstrieren, wie Sie eine logische Befehlszeile über zwei oder mehr Bildschirmzeilen fortsetzen können. Wenn Sie später mit langen Befehlen arbeiten, die über die Breite des Bildschirms hinausgehen, werden Sie diese Methode häufiger einsetzen.

Hinweis – In der Kommando-Shell und der UNIX-Shell von OpenWindows benötigen Sie den Gegenschrägstrich nicht, weil die Befehlszeile dort bei Erreichen des rechten Rands automatisch in der nächsten Zeile fortgesetzt wird.

 2

Eingaben wiederholen

Hinweis – Die in diesem Abschnitt beschriebenen Funktionen stehen nur in der C-Shell zur Verfügung.

Um die letzte Befehlseingabe zu wiederholen, geben Sie `!!` ein und drücken „Return“. Dies ist möglich, weil das System alle Eingaben protokolliert. Um den zuvor eingegebenen Befehl `date` noch einmal an das System zu senden, würden Sie also folgendes eingeben:

```
beispiel% !!
date
Montag, 3. Februar 1992, 10:12:51 Uhr MET
beispiel%
```

Das Befehlsprotokoll wird als *Historie* bezeichnet. Sie können jeden Befehl daraus durch Eingabe von `!`*x*, wiederholen, wobei Sie *x* durch die Nummer des gewünschten Befehls ersetzen. Um den Inhalt der Historie anzuzeigen, geben Sie `history` ein und drücken „Return“. Zum Beispiel:

```
beispiel% history
1   pwd
2   clear
3   ls -l
4   cd /usr/home/worker
5   logname
6   date
7   history
```

Sie können nach dem Ausrufezeichen auch eine negative Zahl angeben. In diesem Fall zählt das System um die entsprechende Anzahl Einträge nach oben und wiederholt den dort stehenden Befehl. Um den Befehl `logname` aus der oben gezeigten Historie zu wiederholen, würden Sie also folgendes eingeben:

```
beispiel% !-2
logname
nina
beispiel%
```

Sie können dem Ausrufezeichen auch die Anfangsbuchstaben eines bereits eingegebenen Befehls nachstellen. Wenn Sie zuvor den Bildschirm schon einmal mit dem Befehl `clear` geleert haben, könnten Sie dies also durch Eingabe von `!cl` wiederholen. Bei dieser Methode müssen Sie jedoch darauf achten, daß Ihre Eingabe eindeutig ist. Wenn mehrere Befehle in der Historie mit den von Ihnen angegebenen Buchstaben beginnen, wird der zuletzt eingegebene Befehl wiederholt.

Befehlsoptionen

Bei vielen Befehlen läßt sich die Ausführung durch Eingabe von *Optionen* beeinflussen. Der Befehl `date` kann beispielsweise mit der Option `-u` eingegeben werden und zeigt dann statt der lokalen Uhrzeit Greenwich Mean Time an:

```
$ date -u
Mo Mär 15 12:18:36 GMT 1993
$
```

Befehlsoptionen bestehen meist aus einem Bindestrich und einem einzelnen Buchstaben. Manche Befehle unterstützen mehrere Optionen, andere überhaupt keine. Wenn Sie einen Befehl mit mehreren Optionen verwenden, können Sie diese entweder separat (`-a -b`) oder zusammen (`-ab`) eingeben.

Befehlsausgabe umleiten

Die von einem Befehl ausgegebenen Daten werden normalerweise auf dem Bildschirm angezeigt. Mit Hilfe spezieller Symbole können Sie die Befehls-ausgabe *umleiten*. Wenn Sie die Ausgabe beispielsweise in einer Datei speichern wollen, verwenden Sie das Größer-Zeichen (>):

```
$ date > beispiel.dat
$
```

In diesem Beispiel wird die Ausgabe des Befehls `date` in eine neue Datei namens `beispiel.dat` umgeleitet, deren Inhalt Sie dann mit dem Befehl `more` anzeigen können:

```
$ more beispiel.dat
Mo Mär 15 12:18:36 GMT 1993
$
```

Weitere Informationen zu dem Befehl `more` finden Sie in Kapitel 3, „Dateien und Verzeichnisse“.

Wenn Sie die Ausgabe eines Befehls als Eingabe an einen anderen Befehl übergeben wollen, verbinden Sie die beiden Befehle zu einer *Pipeline*. Als Umleitungszeichen verwenden Sie in diesem Fall den senkrechten Strich (`|`), der auch als *Pipe* bezeichnet wird.

Auf diese Weise könnten Sie die Ausgabe beispielsweise an den Druckbefehl `lp` übergeben. Um die von dem Befehl `date` ausgegebenen Daten direkt an den Drucker zu senden, würden Sie also folgendes eingeben:

```
$ date | lp
$
```

Weitere Informationen zu dem Befehl `lp` finden Sie im Abschnitt „Anforderungen an den Standarddrucker senden“ in Kapitel 8, „Drucken“.

Die hier vorgestellten Beispiele zur Verwendung von Pipes und zur Befehlsumleitung sind relativ einfach, aber wenn Sie später mit komplexeren Befehlen arbeiten, werden Sie feststellen, daß diese Funktionen sehr vielseitig eingesetzt werden können.

Befehle im Hintergrund ausführen

Befehle, mit denen das System längere Zeit beschäftigt ist, sollten möglichst in den *Hintergrund* gestellt werden. Wenn Sie das nicht tun, erscheint die Eingabeaufforderung nämlich erst wieder, wenn der Befehl vollständig ausgeführt wurde, d. h. es kann unter Umständen einige Zeit dauern, bis Sie weitere Befehle eingeben können.

Um einen Befehl in den Hintergrund zu stellen, geben Sie nach dem Befehlswort das Et-Zeichen & ein. Das System gibt dann die Nummer des Hintergrundprozesses aus. Zum Beispiel:

```
$ bigjob &
[1] 21414
$
```

Der Befehl `bigjob` wird nun im Hintergrund ausgeführt, und Sie können weitere Befehle eingeben. Sobald der Befehl abgearbeitet ist, informiert Sie das System durch eine Statusmeldung. Diese Meldung erscheint zusammen mit der Ausgabe des ersten Befehls, den Sie nach Beendigung des Hintergrund-prozesses eingeben (im folgenden Beispiel ist das der Befehl `date`):

```
$ date
Mo Mär 15 12:18:36 GMT 1993
[1]  + Done    bigjob
$
```

Mit dem Befehl `nohup` (no hangup) können Sie das System anweisen, einen Befehl in jedem Fall vollständig abzuarbeiten. Wenn Sie dies nicht tun, wird der Hintergrundprozeß abgebrochen, sobald Sie sich beim System abmelden.

```
$ nohup bigjob &
[1] 21414
$
```

 2

Hilfe zu SunOS-Befehlen

In diesem Abschnitt werden verschiedene Funktionen beschrieben, mit denen Sie Informationen zu SunOS-Befehlen abrufen können.

Hinweis – Diese Funktionen stehen *zusätzlich* zu den von OpenWindows bereitgestellten Hilfe-Funktionen zur Verfügung.

`man` *pages (Online-Dokumentation zu SunOS-Befehlen) anzeigen*

Wenn Sie den Namen eines Befehls kennen, aber nicht sicher sind, wofür er verwendet wird, können Sie mit dem Befehl `man` ein Online-Dokument mit Informationen zu dem gesuchten Befehl anzeigen. Zum Beispiel:

```
$ man man
```

Nach Eingabe dieses Befehls erscheint das Online-Dokument (`man` page) zu dem Befehl `man` im Anzeigebereich des Fensters. Es enthält eine Liste der verfügbaren Befehlsoptionen und zeigt, in welchem Format der Befehl eingegeben wird. In vielen Online-Dokumenten sind auch Beispiele zur Verwendung des Befehls angegeben. Mit der Leertaste können Sie die nächste Seite aufrufen. Um die Anzeige zu beenden und zur Eingabeaufforderung zurückzukehren, drücken Sie die Taste „Q“.

Kurzbeschreibungen zu SunOS-Befehlen anzeigen (`whatis`)

Mit dem Befehl `whatis` zeigen Sie eine einzeilige Beschreibung der Befehlsfunktion an. Zum Beispiel:

```
$ whatis date
date (1)           -display or set the date
$
```

Die in Klammern stehende Zahl gibt an, zu welcher Gruppe der Befehl gehört. SunOS-Befehle sind in Funktionsgruppen unterteilt. Die meisten Benutzer-befehle gehören zur Gruppe 1. Wenn Sie die Online-Dokumente einer Gruppe ausdrucken, werden sie in alphabetischer Reihenfolge ausgegeben.

Stichwortsuche (`apropos`)

Wenn Sie nicht mehr wissen, mit welchem Befehl eine bestimmte Aktion ausgeführt wird, können Sie mit `apropos` eine Liste aller Befehle abrufen, in deren Kurzbeschreibung ein bestimmtes, von Ihnen angegebenes Stichwort vorkommt. Dabei sollten Sie ein möglichst eindeutiges Stichwort verwenden, da die Liste sonst sehr lang werden kann.

Probieren Sie diesen Befehl mit einem oder mehreren der folgenden Beispiele aus:

- `apropos who`
- `apropos execute`
- `apropos apropos`

Wenn Sie ein sehr allgemeines Stichwort eingegeben haben und eine zu lange Liste erhalten, können Sie die Ausgabe mit „Ctrl-C" abbrechen.

Dateien und Verzeichnisse 3 

SunOS bietet zahlreiche Befehle, mit denen Sie Dateien und Verzeichnisse bearbeiten können. Diese Befehle werden zusammen mit den Datei- oder Verzeichnisnamen in die Befehlszeile eingegeben. In der OpenWindows-Umgebung arbeiten Sie dagegen mit Symbolen, die angeklickt und verschoben werden, und wählen die Dateibefehle aus Menüs.

In diesem Kapitel erfahren Sie, wie Sie Dateien und Verzeichnisse über die SunOS-Befehlszeile manipulieren. Die hier vorgestellten Befehle und Prozeduren sind für eine effiziente Nutzung des Betriebssystems SunOS sehr wichtig. Sie gelten für alle SunOS Befehlszeilen, ob Sie nun unter OpenWindows in einer UNIX- oder Kommando-Shell arbeiten oder sich von einem entfernten Terminal aus beim System anmelden.

Grundlagen

Dateien sind die wichtigsten Datenelemente unter SunOS . Fast jede Datenmenge wird als Datei behandelt:

- **Dokumente** – Dazu gehören Textdateien wie z. B. Briefe oder Berichte, Quellcodes von Programmen und alles, was sonst noch geschrieben und gespeichert wird.
- **Befehle** – Die meisten Befehle sind *ausführbare* Dateien, d. h. sie lösen eine bestimmte Aktion aus. Der Befehl `date`, den Sie im vorigen Kapitel kennengelernt haben, führt beispielsweise ein Programm aus, das die aktuelle Uhrzeit und das aktuelle Datum anzeigt.

- **Geräte** – Auch Peripheriegeräte wie Terminals, Drucker und Laufwerke werden vom Betriebssystem wie Dateien behandelt.
- **Verzeichnisse** – Ein Verzeichnis ist eine Datei, die weitere Dateien enthält.

In den folgenden Abschnitten werden Befehle beschrieben, mit denen Sie Dateien erzeugen, anzeigen, kopieren, bewegen und löschen und den Typ einer Datei ermitteln können.

Dateibefehle

Zu jedem der nachfolgend beschriebenen Befehle gehört ein Beispiel, das Sie an Ihrer Workstation oder Ihrem Terminal nachvollziehen sollten. Die Funktion des Befehls wird dadurch deutlicher, und Sie können sich später besser an ihn erinnern.

Vorbereitungen

Bevor Sie Dateibefehle ausführen, sollten Sie sich vergewissern, daß Sie sich in Ihrem *Home-Verzeichnis* befinden. Dieses Verzeichnis wird bei der Einrichtung Ihrer Zugangsberechtigung vom Systemverwalter angelegt und ist ausschließlich für Ihre Arbeitsdateien vorgesehen. Wenn Sie die Beispiele in diesem Kapitel in Ihrem Home-Verzeichnis ausführen, besteht weniger Gefahr, daß Sie Dateien erzeugen, kopieren, bewegen oder gar löschen, auf die auch von anderen Benutzern zugegriffen wird.

Um in Ihr Home-Verzeichnis zu wechseln, geben Sie den Befehl `cd` ohne weitere Argumente ein. Mit dem Befehl `pwd` können Sie prüfen, in welchem Verzeichnis Sie sich gerade befinden:

```
$ cd
$ pwd
/export/home/benutzername
```

In diesem Beispiel lautet der Pfadname des Home-Verzeichnisses `/export/home/`*benutzername*, wobei *benutzername* für den Namen des Verzeichnis-Eigentümers steht.

Testdatei erzeugen

Mit dem Befehl `touch` erstellen Sie eine leere Datei. Existiert bereits eine Datei mit dem angegebenen Namen, werden lediglich das Zugriffsdatum und die Zugriffszeit dieser Datei aktualisiert.

```
$ touch testdatei
$
```

Dateien anzeigen (`ls`)

Mit dem Befehl `ls` können Sie prüfen, ob die Datei tatsächlich erzeugt wurde:

```
$ ls testdatei
testdatei
```

Wenn Sie `ls` ohne Argumente eingeben, wird der gesamte Inhalt des aktuellen Verzeichnisses angezeigt. Um zu prüfen, ob eine bestimmte Datei existiert, geben Sie `ls` mit dem Namen dieser Datei ein.

Weitere Informationen zu `ls` finden Sie in den *man Pages(1): User Commands*.

Dateien kopieren (`cp`)

Kopieren Sie `testdatei` mit dem Befehl `cp` in eine neue Datei namens `testkopie`:

```
$ cp testdatei testkopie
$
```

Da beide Dateinamen mit den Zeichen „test“ beginnen, können Sie die Originaldatei und die Kopie mit Hilfe eines *Metazeichens* zusammen auflisten. Das Metazeichen * steht für eine beliebige Zeichenfolge. Mit dem Befehl `ls test*` zeigen Sie also alle Dateien in Ihrem Home-Verzeichnis an, deren Name mit „`test`“ beginnt:

```
$ ls test*
testdatei    testkopie
```

Die Datei `testdatei` erscheint an erster Stelle, da das System alle passenden Dateinamen in alphabetischer Reihenfolge anzeigt.

Ausführliche Informationen zu dem Befehl `cp` finden Sie in den *man Pages(1): User Commands*.

Dateien übertragen und umbenennen (`mv`)

Mit dem Befehl `mv` können Sie Dateien umbenennen oder in ein anderes Verzeichnis übertragen. Im folgenden Beispiel benennen Sie `testdatei` in `testfile` um:

```
$ mv testdatei testfile
$
```

Prüfen Sie, ob `testdatei` tatsächlich durch `testfile` ersetzt wurde:

```
$ ls test*
testfile    testkopie
```

Die Datei `testdatei` ist nicht mehr vorhanden.

Weitere Informationen zu dem Befehl `mv` finden Sie in den *man Pages(1): User Commands*.

Dateien löschen (`rm`)

Im letzten Beispiel dieser Gruppe löschen Sie die Datei `testkopie` mit dem Befehl `rm` und überprüfen den Vorgang mit `ls`:

```
$ rm testkopie
$ ls test*
testfile
```



Vorsicht – Eine gelöschte Datei ist unwiderruflich verloren. Wenn Sie keine Sicherungskopie erstellt haben, kann sie nicht mehr wiederhergestellt werden. Sie sollten den Befehl `rm` daher mit Vorsicht verwenden, vor allem dann, wenn Sie mit dem Metazeichen * gleich mehrere Dateien löschen.

Ausführliche Informationen zu `rm` finden Sie in den *man Pages(1): User Commands*.

Inhalt einer Datei anzeigen (`more`, `cat`)

Mit dem Befehl `more` zeigen Sie den Inhalt einer Datei an, deren Namen Sie in der Befehlszeile angeben. Passen nicht alle Daten auf den Bildschirm, erscheint folgende Meldung:

```
--Weiter--(nn%)  [Press space to continue, 'q' to quit.]
```

nn steht für den Prozentsatz des Dateiinhalts, der bereits angezeigt wird.

Sie können den Inhalt einer Datei auch mit dem Befehl `cat` anzeigen. In diesem Fall besteht jedoch keine Möglichkeit, die Anzeige anzuhalten. Dieser Befehl wird daher vor allem für die Verknüpfung von zwei oder mehr Dateien zu einer größeren Datei verwendet. Zum Beispiel:

```
$ cat datei1 datei2 datei3 > gesamtdatei
$ ls *datei
gesamtdatei
datei1
datei2
datei3
$
```

Weitere Informationen zu den Befehlen `more` und `cat` finden Sie in den *man Pages(1): User Commands*.

Dateityp anzeigen (`file`)

Manche Dateien, z. B. Binärdateien und ausführbare Dateien, können weder gedruckt noch auf dem Bildschirm angezeigt werden. Wenn Sie sich über den Typ einer Datei nicht im klaren sind, können Sie ihn mit dem Befehl `file` abrufen:

```
$ file testfile
testfile:    ASCII-Text
```

Verzeichnisbefehle

Sie wissen jetzt, wie man Dateien auflistet, kopiert, umbenennt und löscht. Wichtig ist aber auch, daß Sie diese Dateien möglichst rasch finden. Deshalb muß das Dateisystem auf eine logische und sinnvolle Weise strukturiert sein. Diese Strukturierung, die über Verzeichnisse erreicht wird, bildet das Thema der folgenden Abschnitte.

Verzeichnishierarchie

Dateien werden in Verzeichnissen abgelegt, die wiederum Teil anderer, übergeordneter Verzeichnisse sind. Daraus ergibt sich eine Hierarchie, an deren Spitze das Root-Verzeichnis steht. Dieses Verzeichnis wird durch das Zeichen `/` repräsentiert.

Abbildung 3-1 zeigt, daß jedes Verzeichnis weitere Verzeichnisse enthalten kann. Die einzelnen Verzeichnisebenen werden ebenfalls durch das Zeichen / getrennt. Das Root-Verzeichnis `/` enthält unter anderem die Unterverzeichnisse `/usr`, `/bin`, `/home` und `/lib`. Das Unterverzeichnis `/home` enthält die Arbeitsverzeichnisse der Benutzer: `/user1`, `/user2` und `/user3`.

Um auf ein Verzeichnis (und die darin enthaltenen Dateien) zu verweisen, geben Sie die Namen der direkt übergeordneten Verzeichnisse an. Ein solcher Verweis wird als *Pfadname* bezeichnet. Der Pfadname zu dem Verzeichnis `/user3` lautet also `/home/user3`. Der erste Schrägstrich steht für das Root-Verzeichnis, danach folgt der Name des übergeordneten Verzeichnisses `home`, ein zweiter Schrägstrich trennt diese Verzeichnisebene von der nächsten, und zum Schluß kommt der Name des gesuchten Verzeichnisses, `user3`.

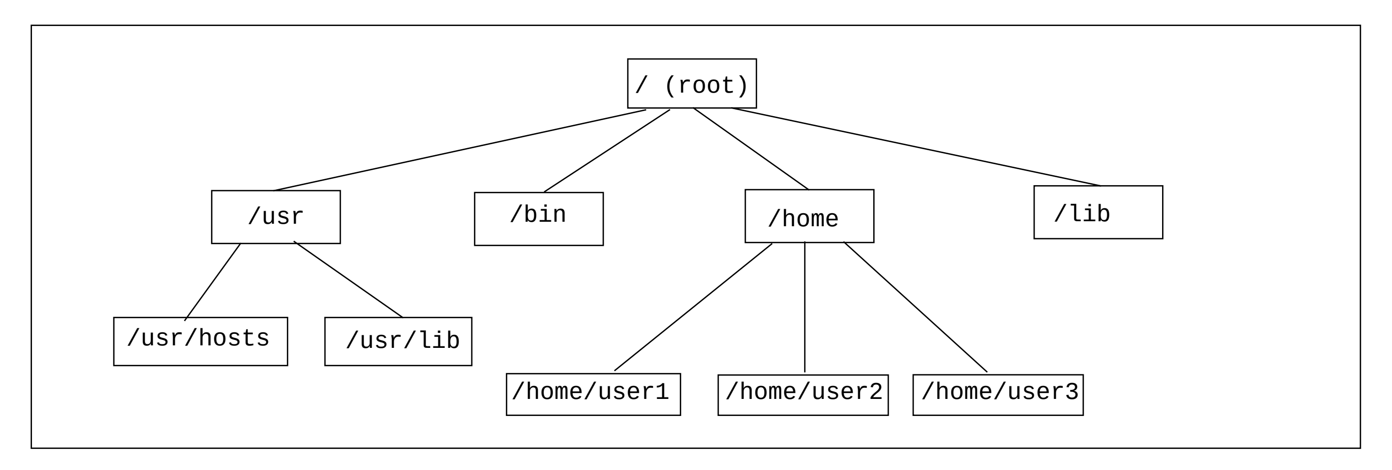


Abbildung 3-1 Struktur des Dateisystems

Innerhalb eines Verzeichnisses darf jeder Dateiname (dazu zählen auch die Namen der Unterverzeichnisse) nur einmal vorkommen. In anderen Verzeichnissen kann er sich dagegen wiederholen. Die Verzeichnisse `/usr/lib` und `/lib` in unserem Beispiel können also nebeneinander existieren, weil sie unterschiedliche Pfadnamen haben.

Der Standort einer Datei innerhalb des Dateisystems wird ebenfalls über einen Pfadnamen angegeben. Nehmen wir an, das Verzeichnis `/home/user2` enthält eine Datei namens `bericht5`. Der Pfadname dieser Datei lautet demnach `/home/user2/bericht5`, d. h. die Datei befindet sich im Verzeichnis `user2`, einem Unterverzeichnis von `home`, das wiederum ein Unterverzeichnis des Root-Verzeichnisses (`/`) ist.

Verzeichnisse können sowohl Unterverzeichnisse als auch Dateien enthalten.

Aktuelles Verzeichnis anzeigen (`pwd`*)*

Mit dem Befehl `pwd` können Sie feststellen, an welcher Stelle innerhalb der Verzeichnisstruktur Sie sich gerade befinden. Dieses Verzeichnis wird als *aktuelles Verzeichnis* oder *Arbeitsverzeichnis* bezeichnet. Zum Beispiel:

```
$ pwd
/home/user1
```

Home-Verzeichnis

Der Systemverwalter legt für jeden Benutzer ein *Home-Verzeichnis* an, in dem die Arbeitsdateien dieses Benutzers gespeichert werden. Wenn Sie OpenWindows starten und eine Kommando- oder UNIX-Shell öffnen, befinden Sie sich automatisch in Ihrem Home-Verzeichnis.

Aktuelles Verzeichnis wechseln (cd)

Mit dem Befehl `cd` wechseln Sie in ein anderes Verzeichnis. Zum Beispiel:

```
$ cd /usr/lib
$ pwd
/usr/lib
```

Wenn Sie `cd` ohne Verzeichnisnamen eingeben, kehren Sie in Ihr Home-Verzeichnis zurück. Im folgenden Beispiel heißt das Home-Verzeichnis `/home/user1`:

```
$ cd
$ pwd
/home/user1
```

In der C-Shell können Sie das Zeichen ~ (Tilde) als Abkürzung für Ihr Home-Verzeichnis verwenden, statt den vollständigen Pfadnamen einzugeben. Mit folgendem Befehl würden Sie also in das Unterverzeichnis `musik` in Ihrem Home-Verzeichnis wechseln:

```
beispiel% cd ~/musik
```

Sie können mit der Tilde auch das Home-Verzeichnis eines anderen Benutzers angeben. Zum Beispiel:

```
beispiel% cd ~benutzername
```

Dabei ersetzen Sie *benutzername* durch den Login-Namen des Benutzers, zu dessen Home-Verzeichnis Sie wechseln wollen.

Hinweis – In der Bourne-Shell ist die Verwendung von ~ als Abkürzung für das Home-Verzeichnis nicht möglich.

Wenn Sie mit einer Bourne-Shell arbeiten und Ihr Systemverwalter das System entsprechend konfiguriert hat, können Sie mit `$home` das Home-Verzeichnis angeben. Zum Beispiel:

```
$ $home/musik
```

Auch in diesem Fall können Sie sowohl das eigene Home-Verzeichnis wie auch die Home-Verzeichnisse anderer Benutzer angeben:

```
$ $home benutzername
```

benutzername wird durch den Login-Namen des Benutzers ersetzt, dem das Verzeichnis gehört.

Das einem Verzeichnis direkt übergeordnete Verzeichnis kann auch mit dem Symbol `..` („Punkt-Punkt") angegeben werden. Im folgenden Beispiel wechseln Sie aus dem Verzeichnis `/home/user1` in das darüberliegende Verzeichnis `/home`:

```
$ pwd
/home/user1
$ cd ..
$ pwd
/home
```

Nehmen wir an, Ihr Home-Verzeichnis heißt `/home/user1`, und Sie möchten auf Dateien im Verzeichnis `/home/user2` zugreifen. Um in dieses Verzeichnis zu wechseln, geben Sie einfach folgendes ein (den Befehl `pwd` können Sie auch weglassen, er soll nur zeigen, wo Sie sich vor und nach der Eingabe von `cd ../user2` befinden):

```
$ pwd
/home/user1
$ cd ../user2
$ pwd
/home/user2
```

Bei dieser Methode brauchen Sie also nicht den gesamten Pfadnamen (`/home/user2`) einzugeben.

Verzeichnisse anlegen (`mkdir`)

Um ein neues Verzeichnis anzulegen, geben Sie den Befehl `mkdir` und den gewünschten Verzeichnisnamen ein:

```
$ mkdir obst
$ cd obst
$ mkdir aprikose
$ cd aprikose
$ pwd
/home/user2/obst/aprikose
```

Relative Pfadnamen

Der vollständige Pfadname eines Verzeichnisses oder einer Datei beschreibt die gesamte Verzeichnisstruktur zwischen diesem Verzeichnis oder dieser Datei und dem Root-Verzeichnis (`/`). Häufig genügt es jedoch, den Pfadnamen mit dem aktuellen Verzeichnis beginnen zu lassen. Ein solcher Pfadname wird als *relativer* Pfadname bezeichnet, weil er den Standort des Verzeichnisses (oder der Datei) relativ zum Arbeitsverzeichnis angibt.

Wenn Sie in ein Unterverzeichnis wechseln wollen, brauchen Sie nicht den vollständigen Pfadnamen, sondern nur den Verzeichnisnamen anzugeben. Im vorigen Beispiel wurde in dem Befehl `cd obst` der relative Pfadname des Verzeichnisses `obst` verwendet. Der vollständige Pfadname würde dagegen `/home/user2/obst` lauten.

Legen Sie in Ihrem Home-Verzeichnis weitere Verzeichnisse und Unterverzeichnisse an, und bewegen Sie sich mit Hilfe relativer und vollständiger Pfadnamen durch die Verzeichnisstruktur. Überprüfen Sie dabei jedesmal Ihren aktuellen Standort mit dem Befehl `pwd`.

Verzeichnisse übertragen und umbenennen (`mv`)

Um ein Verzeichnis umzubenennen, „übertragen" Sie es mit dem Befehl `mv` in einen anderen Namen:

```
$ pwd
/home/user2/obst
$ ls
aprikose
$ mv aprikose mango
$ ls
mango
```

Mit diesem Befehl können Sie ein Verzeichnis auch in ein anderes Verzeichnis übertragen:

```
$ pwd
/home/user2/obst
$ ls
mango
$ cd ..
$ mkdir obst2
$ cd obst
$ mv mango ../obst2
$ ls ../obst2
mango
```

In diesem Beispiel erstellen Sie ein neues Verzeichnis namens `obst2` und übertragen das Unterverzeichnis `mango` mit `mv` aus dem Verzeichnis `obst` in das neue Verzeichnis.

Verzeichnisse kopieren (`cp -r`)

Mit dem Befehl `cp -r` kopieren Sie ein Verzeichnis und die darin enthaltenen Dateien:

```
$ cp -r obst obst3
$
```

In diesem Beispiel kopieren Sie alle Dateien und Unterverzeichnisse im Verzeichnis `obst` in ein neues Verzeichnis namens `obst3`. Die Option `-r` gibt an, daß es sich um eine *rekursive* Kopie handelt. Wenn Sie versuchen, ein Verzeichnis ohne diese Option zu kopieren, erhalten Sie eine Fehlermeldung.

Verzeichnisse löschen (`rmdir`)

Um ein leeres Verzeichnis zu löschen, geben Sie den Befehl `rmdir` und den Verzeichnisnamen ein:

```
$ rmdir obst3
$
```

Enthält das Verzeichnis noch Dateien oder Unterverzeichnisse, wird es nicht gelöscht.

Wenn Sie ein Verzeichnis mit seinem gesamten Inhalt löschen wollen, geben Sie den Befehl `rm` mit der Option `-r` (für *rekursiv*) ein. Zum Beispiel:

```
$ rm -r obst3
$
```



Vorsicht – Verzeichnisse, die Sie mit dem Befehl `rmdir` löschen, und mit `rm -r` gelöschte Verzeichnisinhalte können *nicht* wiederhergestellt werden.

Zwei Dateien vergleichen (`diff`)

Dateien werden häufig von mehreren Benutzern kopiert und dann individuell bearbeitet. Mit dem Befehl `diff` können Sie feststellen, ob und wie sich zwei Versionen einer ASCII-Textdatei unterscheiden:

```
$ diff linkedatei rechtedatei
```

Daraufhin werden nacheinander alle Zeilen in `linkedatei` und `rechtedatei` miteinander verglichen. Stellt das System fest, das sich ein oder mehrere Zeilenpaare unterscheiden, prüft es, ob Teile der Zeile hinzugefügt, gelöscht oder verändert wurden und welche Zeilen davon betroffen sind. Ausgegeben werden die Nummer(n) der entsprechenden Zeile(n) in beiden Dateien und der differierende Text.

Wurde Text hinzugefügt, zeigt `diff` eine Zeile in folgendem Format an:

l[,*l*] a *r*[,*r*]

l steht für eine Zeilennummer in `linkedatei` und *r* für eine Zeilennummer in `rechtedatei`.

Wurde Text gelöscht, zeigt `diff` statt dem `a` ein `d` an. Beruht der Unterschied auf einer Änderung, wird ein `c` angezeigt.

Der Text aus beiden Dateien folgt unmittelbar auf die Zeilennummer. Der Text aus `linkedatei` ist durch das Zeichen < und der Text aus `rechtedatei` durch das Zeichen > gekennzeichnet.

Im nächsten Beispiel wird der Inhalt zweier Dateien zunächst angezeigt und danach mit `diff` verglichen:

```
$ cat termine.7.15
Woche vom 7/15

Tag:  Zeit:        Termin:              Kommentar:

D     10:00        Hardware (Bspr.)     alle zwei Wochen
M     1:30         Software (Bspr.)
D     3:00         Doku-Meeting
F     1:00         Personalgespräch
$ cat termine.7.22
Woche von 7/22

Tag:  Zeit:        Terminliste:         Kommentar:

M     8:30         Mitarbeiter-Bespr.   ganztägig
D     10:00        Hardware (Bspr.)     alle zwei Wochen
M     1:30         Software (Bspr.)
D     3:00         Doku-Meeting
$ diff termine.7.15 termine.7.22
1c1
< Woche von 7/15
---
> Woche von 7/22
4a5
> M     8:30         Mitarbeiter-Bespr.   ganztägig
8d8
< F     1:00         Personalgespräch
```

Sind die beiden Dateien identisch, werden keine Daten ausgegeben.

Der Befehl `diff` kann mit zahlreichen Optionen eingegeben werden. Näheres dazu finden Sie in den *man Pages(1): User Commands*.

Drei Dateien vergleichen (`diff3`)

Wenn Sie drei Versionen einer Datei gleichzeitig vergleichen möchten, verwenden Sie den Befehl `diff3`:

```
$ diff3 datei1 datei2 datei3
```

`diff3` vergleicht die Versionen und zeigt differierenden Text mit folgenden Symbolen an:

==== Abweichungen in allen Dateien

====1 Abweichungen in *datei1*

====2 Abweichungen in *datei 2*

====3 Abweichungen in *datei 3*

Große Dateien vergleichen (`bdiff`)

Wenn die zu vergleichenden Dateien sehr umfangreich sind, verwenden Sie statt `diff` den Befehl `bdiff`. Die Eingabesyntax ist gleich:

```
$ bdiff linkedatei rechtedatei
```

`bdiff` sollte verwendet werden, wenn die Dateien mehr als etwa 3500 Zeilen Text enthalten.

Dateien suchen (`find`)

Mit dem Befehl `find` können Sie in einem Verzeichnis und allen darin enthaltenen Unterverzeichnissen nach Dateien suchen, die bestimmte Kriterien erfüllen. Das könnten beispielsweise Dateien sein, deren Namen eine bestimmte Zeichenfolge enthalten oder die innerhalb eines bestimmten Zeitraums bearbeitet wurden.

Im Gegensatz zu den meisten anderen Befehlen muß bei `find` der Verzeichnisname vor den Befehlsoptionen stehen:

```
$ find verzeichnis optionen
```

verzeichnis ist der Name des Verzeichnisses, in dem gesucht werden soll. *optionen* wird durch eine oder mehrere Befehlsoptionen ersetzt, die bei diesem Befehl ausnahmsweise mehrere Zeichen lang sind.

Jede Option beschreibt ein Auswahlkriterium. Damit eine Datei in die Liste der gefundenen Dateien aufgenommen wird, muß sie alle Kriterien erfüllen. Je mehr Kriterien Sie also angeben, desto enger wird das Feld der Dateien, auf die Ihre Beschreibung zutrifft. Mit der Option `-print` legen Sie fest, daß die Ergebnisse auf dem Bildschirm angezeigt werden sollen. (Dies muß explizit angegeben werden, weil Sie mit `find` auch andere Befehle ausführen lassen können und die Ausgabe in diesem Fall häufig überflüssig ist.)

Mit der Option `-name` *dateiname* suchen Sie nach allen Dateien mit der Zeichenfolge *dateiname* als letztem Element im Pfadnamen. Das letzte Element im Pfadnamen der Datei `/usr/lib/calendar` ist beispielsweise `calendar`. Dieser Teil des Pfadnamens wird auch als *Basisname* bezeichnet.

Im folgenden Beispiel zeigen Sie alle Dateien und Verzeichnisse im aktuellen Verzeichnis an, deren Namen mit dem Buchstaben „`s`" enden:

```
$ find . -name '*s' -print
./programs
./programs/graphics
./programs/graphics/gks
./src/gks
$
```

Wenn Sie in *dateiname* ein Metazeichen verwenden, muß die Zeichenfolge in einfache Anführungszeichen gesetzt werden.

Weitere Optionen:

`-user` *benutzer_id*

Sucht nach allen Dateien, die einem bestimmten Benutzer gehören. Ersetzen Sie *benutzer_id* durch den Login-Namen oder die Benutzernummer.

`-group` *gruppe*

Sucht nach allen Dateien, die der angegebenen Benutzergruppe gehören.

`-m time` *n*

Sucht nach allen Dateien, die innerhalb der letzten *n* Tage geändert wurden.

`-newer` *ref_datei*

Sucht nach allen Dateien, die nach *ref_datei* geändert wurden.

Wenn Sie die Suchkriterien nach Priorität ordnen wollen, können Sie sie im Format `\(`*optionen* `\)` eingeben. Innerhalb der Klammern können Sie dann die Option `-o` verwenden. Steht diese Option zwischen zwei Kriterien, werden auch Dateien ausgewählt, die nur eine der beiden Kriterien erfüllen:

```
$ find . \( -name AAA -o -name BBB \) -print
./AAA
./BBB
```

Soll nach allen Dateien gesucht werden, die den angegebenen Kriterien *nicht* entsprechen, stellen Sie den Optionen die Zeichenfolge `\!` voran:

```
$ find . \!-name BBB -print
./AAA
```

Mit folgender Option können Sie außerdem festlegen, daß auf die gefundenen Dateien ein Befehl angewendet werden soll:

`-exec` *befehl* `'{}' \;`

Die in Anführungszeichen stehenden geschweiften Klammern stehen dabei für die von `find` ausgewählten Dateien.

Mit dieser Methode können Sie beispielsweise Arbeitsdateien, die nur vorübergehend benötigt werden, automatisch löschen lassen. Voraussetzung ist, daß Sie solchen Dateien immer die gleichen Namen zuweisen. Mit dem folgenden Befehl werden beispielsweise alle Dateien ausgewählt und gelöscht, die `temp` oder `dummy` heißen:

```
$ find . \( -name temp -o -name dummy \) -exec rm '{}' \;
```

Weitere Informationen zu `find` finden Sie in den *man Pages(1): User Commands*.

Zugriffsrechte

Hinweis – Da Sie es bei der täglichen Arbeit häufig mit Zugriffsrechten zu tun haben werden, sollten Sie diesen Abschnitt besonders aufmerksam lesen.

Zugriffsrechte legen fest, wer auf eine Datei oder ein Verzeichnis zugreifen darf und welche Zugriffsarten gestattet sind.

Es gibt drei Arten von Zugriffsrechten:

- `r` – *Leseberechtigung*. Damit Sie eine Datei öffnen oder kopieren bzw. den Inhalt eines Verzeichnisses anzeigen können, müssen Sie für die Datei oder das Verzeichnis leseberechtigt sein.
- `w` – *Schreibberechtigung*. Damit Sie eine Datei ändern, löschen oder umbenennen bzw. den Inhalt eines Verzeichnisses durch Hinzufügen oder Löschen von Dateien ändern können, müssen Sie für die Datei oder das Verzeichnis schreibberechtigt sein.
- `x` – *Ausführberechtigung*. Damit Sie auf eine ausführbare Datei bzw. die Unterverzeichnisse eines Verzeichnisses zugreifen können, müssen Sie für die Datei oder das Verzeichnis ausführberechtigt sein.

Zugriffsrechte können an drei Arten von Benutzern vergeben werden:

- Selbst – Der Eigentümer der Datei oder des Verzeichnisses.
- Gruppe – Alle Benutzer, die der gleichen Benutzergruppe angehören wie der Eigentümer. Benutzergruppen werden vom Systemverwalter eingerichtet und verwaltet.
- Andere – Alle weiteren Benutzer.

Zugriffsrechte und Status anzeigen (`ls -l`)

Daß Sie mit dem Befehl `ls` Dateinamen auflisten können, wissen Sie bereits. Wenn Sie den Befehl mit der Option `-l` eingeben, werden außer den Dateinamen eine Reihe weiterer Dateiattribute angezeigt. Abbildung 3-2 zeigt ein Beispiel für die Ausgabe des Befehls `ls -l`:

```
$ pwd
/home/hostname/user2
$ ls -l
total 8
drwxr-xr-x  2 user2          1024 Feb  9 14:22 verzeichnis1
-rw-r--r--  1 user2             0 Feb 10 10:20 leerdatei
-rw-r--r--  1 user2        104357 Feb  5 08:20 große-datei
drwxr-xr-x  3 user2          1024 Feb 10 11:13 obst2
```

Berechtigungen Links Eigentümer Größe Datum Zeit Name

Abbildung 3-2 Dateiattribute anzeigen

Das erste Zeichen gibt den Dateityp an: Ein Bindestrich (`-`) steht für eine Textdatei und ein `d` für ein Verzeichnis. Darüber hinaus gibt es weitere Dateitypen, die hier jedoch nicht behandelt werden.

Die folgenden neun Zeichen geben an, welche Zugriffsrechte für die Datei oder das Verzeichnis gelten. Diese Zeichen sind in drei Blöcke zu je drei Zeichen gegliedert, wobei der erste Block für den Eigentümer, der zweite für die Gruppe und der dritte für alle anderen Benutzer gilt. Für die Datei `leerdatei` gelten beispielsweise die Berechtigungen `rw-r--r--`, d. h. der Eigentümer der Datei verfügt über Lese- und Schreibberechtigung, während alle anderen Benutzer sie nur lesen, aber nicht verändern dürfen. Der Inhalt des Verzeichnisses `obst2` darf nur vom Eigentümer geändert werden, da alle anderen Benutzer lediglich über Lese- und Ausführberechtigung verfügen.

Außer den Zugriffsrechten werden folgende Informationen angezeigt:

- Anzahl der Verweise (Links) auf die Datei oder das Verzeichnis
- Name des Eigentümers (in diesem Fall `user2`)
- Größe der Datei (in Byte)
- Datum und Zeitpunkt der letzten Änderung
- Datei- oder Verzeichnisname

Wechseln Sie mit `cd` in Ihr Home-Verzeichnis, und probieren Sie den Befehl `ls -l` aus. Auf Ihrem System werden natürlich andere Daten angezeigt als in unserem Beispiel.

Geben Sie den Befehl danach in folgendem Format ein:

```
$ ls -l verzeichnisname
```

Ersetzen Sie *verzeichnisname* dabei durch den Namen eines Verzeichnisses in Ihrem Dateisystem. Der Befehl `ls -l` zeigt daraufhin die Attribute aller Dateien und (falls vorhanden) Unterverzeichnisse in diesem Verzeichnis an.

Versteckte Dateien anzeigen (`ls -a`)

Versteckte Dateien können mit dem Befehl `ls` ohne Optionen nicht angezeigt werden. Diese Dateien heißen auch Punktdateien, weil der Dateiname mit einem Punkt beginnt. Beispiele für Punktdateien sind `.cshrc`, `.login` und `.profile`. Um neben den normalen Dateien auch eventuell vorhandene Punktdateien mit aufzulisten, geben Sie `ls` mit der Option `-a` ein:

```
$ ls -a
.
..
.cshrc
.login
.profile
leerdatei
```

Die Punktdateien stehen vor den normalen Dateien. Die Dateien `.` und `..` repräsentieren das aktuelle Verzeichnis und das übergeordnete Verzeichnis.

Punktdateien werden im allgemeinen nur vom System verwendet und vom Benutzer nicht verändert. Es gibt jedoch Ausnahmen, von denen einige in späteren Kapiteln behandelt werden.

Zugriffsrechte ändern (chmod)

Mit dem Befehl chmod ändern Sie die Zugriffsrechte für eine Datei oder ein Verzeichnis. Dies ist allerdings nur möglich, wenn Sie Eigentümer der Datei sind oder sich als Superuser angemeldet haben. Der Befehl chmod wird in folgendem Format eingegeben:

```
chmod  berechtigungen  name
```

Dabei steht *berechtigungen* für die geänderten Zugriffsrechte und *name* für den Namen der Datei oder des Verzeichnisses.

Zugriffsrechte können auf unterschiedliche Weise definiert werden. Am einfachsten geht es in folgender Form:

1. Verwenden Sie einen oder mehrere der folgenden Buchstaben, um anzugeben, für welche Benutzer die Zugriffsrechte gelten sollen:
   - u (für den Eigentümer)
   - g (für die Gruppe)
   - o (für alle anderen Benutzer)
   - a (für alle drei oben genannten Kategorien)
2. Geben Sie an, ob Berechtigungen hinzugefügt (+) oder entfernt (-) werden.
3. Verwenden Sie einen oder mehrere der folgenden Buchstaben, um die neuen Zugriffsrechte festzulegen:
   - r (für Leseberechtigung)
   - w (für Schreibberechtigung)
   - x (für Ausführberechtigung)

Im folgenden Beispiel erteilt der Eigentümer des Verzeichnisses mango allen Mitgliedern seiner Benutzergruppe Schreibberechtigung für das Verzeichnis (*berechtigungen* wird also durch g+w und *name* durch mango ersetzt):

```
$ ls -l mango
drwxr-xr-x  3 user2          1024 Feb 10 11:15 mango
$ chmod g+w mango
$ ls -l mango
drwxrwxr-x  3 user2          1024 Feb 10 11:15 mango
$
```

Statt des Bindestrichs, der für „keine Schreibberechtigung“ steht, erscheint in den Zugriffsrechten für die Gruppe nun ein w.

Um Lese- und Ausführzugriffe auf das Verzeichnis von Benutzern außerhalb der Gruppe auszuschließen, würden Sie folgendes eingeben:

```
$ ls -l mango
drwxrwxr-x  3 user2           1024 Feb 10 11:15 mango
$ chmod o-rx mango
$ ls -l mango
drwxrwx---  3 user2           1024 Feb 10 11:15 mango
$
```

In den Zugriffsrechten für andere Benutzer werden nun statt dem r (für Leseberechtigung) und dem x (für Ausführberechtigung) zwei Bindestriche angezeigt.

Wenn Sie eine neue Datei oder ein neues Verzeichnis anlegen, vergibt das System automatisch Zugriffsrechte.

Für neue Dateien gelten normalerweise folgende Zugriffsrechte:

```
-rw-r--r--
```

Bei neuen Verzeichnissen werden folgende Zugriffsrechte vergeben:

```
drwxr-xr-x
```

Damit eine neue Datei namens pfirsich von ihrem Eigentümer (user2) ausgeführt werden kann, müßte dieser (oder der Superuser) also zunächst folgendes eingeben:

```
$ ls -l pfirsich
-rw-r--r--  3 user2           1024 Feb 10 12:27 pfirsich
$ chmod u+x pfirsich
$ ls -l pfirsich
-rwxr--r--  3 user2           1024 Feb 10 12:27 pfirsich
$
```

Mit der Option a können Sie Zugriffsrechte gleichzeitig an alle drei Benutzerkategorien vergeben. Um allen Benutzern Ausführberechtigung für eine neue Datei namens orange zu erteilen, würden Sie also folgendes eingeben:

```
$ ls -l orange
-rw-r--r--  3 user2          1024 Feb 10 11:31 orange
$ chmod a+x orange
$ ls -l orange
-rwxr-xr-x  3 user2          1024 Feb 10 11:31 orange
$
```

Nach Ausführung des Befehls chmod erscheint in allen drei Blöcken das Zeichen x.

Zugriffsrechte können mit Hilfe des Metazeichens * auch für mehrere Dateien oder Verzeichnisse gleichzeitig vergeben werden. Im folgenden Beispiel werden die Zugriffsrechte für die Dateien im Verzeichnis obst so geändert, daß nur noch der Eigentümer des Verzeichnisses über Schreibberechtigung verfügt:

```
$ pwd
/home/user2/obst
$ ls -l
-rwxrwxrwx  3 user2         21032 Feb 12 10:31 kirsche
-rwxrwxrwx  2 user2            68 Feb 10 11:09 erdbeere
-rwxrwxrwx  3 user2         12675 Feb 08 09:31 orange
-rwxrwxrwx  1 user2          1024 Feb 14 16:38 zitrone
$ chmod go-w *
$ ls -l
-rwxr-xr-x  3 user2         21032 Feb 12 10:31 kirsche
-rwxr-xr-x  2 user2            68 Feb 10 11:09 erdbeere
-rwxr-xr-x  3 user2         12675 Feb 08 09:31 orange
-rwxr-xr-x  1 user2          1024 Feb 14 16:38 zitrone
$
```

Da der Befehl chmod in der oben gezeigten Form immer auf die Dateien im aktuellen Verzeichnis angewendet wird, sollten Sie vorher mit pwd prüfen, ob Sie sich im richtigen Verzeichnis befinden.

Absolute Zugriffsrechte vergeben

In den bisherigen Beispielen wurden Zugriffsrechte *relativ* zu den aktuell geltenden Einstellungen geändert. Sie können Zugriffsrechte mit `chmod` jedoch auch ändern, ohne die aktuellen Einstellungen zu kennen. Bei der Vergabe solcher *absoluter* Zugriffsrechte werden Benutzerkategorie und Zugriffsberechtigung nicht durch Buchstabenkürzel, sondern durch einen Zahlencode angegeben.

Die Befehlssyntax lautet in diesem Fall:

`chmod` *numcode name*

numcode steht für den Berechtigungscode und *name* für den Namen der Datei oder des Verzeichnisses, deren / dessen Zugriffsrechte geändert werden sollen.

Der Berechtigungscode besteht aus je einer Zahl für die drei Benutzer-kategorien. Mit dem folgenden Befehl erhalten der Eigentümer des Verzeichnisses `orange` und seine Gruppe Lese-, Schreib- und Ausführberechtigung, alle anderen Benutzer dagegen nur Ausführberechtigung:

```
$ chmod 771 orange
```

Tabelle 3-1 zeigt, aus welchen Elementen sich der Berechtigungscode `771` zusammensetzt.

Tabelle 3-1 Zugriffsrechte für `orange`

Berechtigung	Eigentümer	Gruppe	Andere
Lesen	4	4	0
Schreiben	2	2	0
Ausführen	1	1	1
Gesamt	7	7	1

Jede Spalte in Tabelle 3-1 repräsentiert eine Benutzerkategorie. Jeder Berechtigungsart ist ein bestimmter Wert zugeordnet: 4 für Leseberechtigung, 2 für Schreibberechtigung und 1 für Ausführberechtigung. Die Summe der ersten, zweiten und dritten Spalte entspricht der ersten, zweiten und dritten Ziffer des Berechtigungscodes.

Hier ein weiteres Beispiel für die Vergabe von absoluten Zugriffsrechten, bei dem die Ergebnisse mit dem Befehl `ls -l` geprüft werden:

```
$ ls -l zitrone
-rw-r--r--  3 user2          1024 Feb 10 11:46 zitrone
$ chmod 755 zitrone
$ ls -l zitrone
-rwxr-xr-x  3 user2          1024 Feb 10 11:48 zitrone
$
```

Für die Datei `zitrone` gelten also folgende Zugriffsrechte: der Eigentümer kann sie lesen, schreiben und ausführen, während die Gruppe und alle anderen Benutzer nur Lese- und Ausführberechtigung haben. Tabelle 3-2 zeigt, wie der Berechtigungscode ermittelt wird.

Tabelle 3-2 Zugriffsrechte für `zitrone`

Berechti-gung	Eigen-tümer	Gruppe	Andere
Lesen	4	4	4
Schreiben	2	0	0
Ausführen	1	1	1
Gesamt	7	5	5

Um als Eigentümer der Datei `kirsche` sich selbst, der Gruppe und allen anderen Benutzern Lese-, Schreib- und Ausführberechtigung zu erteilen, würden Sie also folgendes eingeben:

```
$ ls -l kirsche
-rw-r--r--  3 user2          1024 Feb 10 11:51 kirsche
$ chmod 777 kirsche
$ ls -l kirsche
-rwxrwxrwx  3 user2          1024 Feb 10 11:53 kirsche
$
```

Tabelle 3-3 zeigt, wie dieser Code zustandekommt:

Tabelle 3-3 Zugriffsrechte für `kirsche`

Berechti-gung	Eigen-tümer	Gruppe	Andere
Lesen	4	4	4
Schreiben	2	2	2
Ausführen	1	1	1
Gesamt	7	7	7

Der Code `777` entspricht also der maximalen Anzahl Berechtigungen, die Sie für eine Datei oder ein Verzeichnis vergeben können.

Auch bei dieser Methode können Sie mit dem Metazeichen * Zugriffsrechte für alle Dateien des aktuellen Verzeichnisses definieren. Im folgenden Beispiel vergibt der Eigentümer des Verzeichnisses `obst` an sich selbst Lese-, Schreib- und Ausführberechtigung für alle darin enthaltenen Dateien, seine Gruppe erhält Lese- und Ausführberechtigung und alle anderen Benutzer lediglich Ausführberechtigung:

```
$ pwd
/home/user2/obst
$ ls -l
-rwxrwxrwx  3 user2          21032 Feb 12 10:31 kirsche
-rwxrwxrwx  2 user2             68 Feb 10 11:09 erdbeere
-rwxrwxrwx  3 user2          12675 Feb 08 09:31 orange
-rwxrwxrwx  1 user2           1024 Feb 14 16:38 zitrone
$ chmod 751 *
$ ls -l
-rwxr-x--x  3 user2          21032 Feb 12 10:31 kirsche
-rwxr-x--x  2 user2             68 Feb 10 11:09 erdbeere
-rwxr-x--x  3 user2          12675 Feb 08 09:31 orange
-rwxr-x--x  1 user2           1024 Feb 14 16:38 zitrone
$
```

Mit dem Befehl `pwd` wird zu Beginn geprüft, ob `obst` tatsächlich aktuelles Verzeichnis ist. Der Befehl `ls -l` wird hier, wie auch in den vorhergehenden Beispielen, nur verwendet, um das Ergebnis der Änderung zu verdeutlichen.

Weitere Informationen zu `chmod` siehe *man Pages(1): User Commands*.

Suchen in Dateien 4



In diesem Kapitel erfahren Sie, wie Dateien und Verzeichnisse nach Stichwörtern durchsuchen.

Nach Zeichenfolgen suchen (`grep`)

Mit dem Befehl `grep` können Sie in einer Datei nach einer bestimmten Zeichenfolge suchen. `grep` wird in folgendem Format eingegeben:

```
$ grep zeichenfolge datei
```

zeichenfolge ist die gesuchte Zeichenfolge und *datei* die Datei, in der gesucht werden soll.

Hinweis – Eine *Zeichenfolge* ist eine beliebige Kombination aus einem oder mehreren Zeichen. Einzelne Buchstaben werden ebenso als Zeichenfolgen bezeichnet wie Wörter oder ganze Sätze. Zeichenfolgen können außerdem Leerzeichen, Satzzeichen und unsichtbare Steuerzeichen enthalten.

Im folgenden Beispiel wird in der Datei `telefon` nach der Rufnummer von Edgar Allan Poe gesucht:

```
$ grep Poe telefon
Edgar Allan Poe     x728364
$
```

Wenn die angegebene Zeichenfolge mehrmals vorkommt, werden alle passenden Zeilen angezeigt:

```
$ grep Edgar telefon
Edgar Winter        x764383
Edgar Allan Poe     x728364
$ grep Ed telefon
Eduard Müller       x742362
Edgar Winter        x764383
Edgar Allan Poe     x728364
$
```

Da `grep` zwischen Groß- und Kleinbuchstaben unterscheidet, müssen Sie bei der Eingabe auf die korrekte Schreibweise achten:

```
$ grep edgar telefon
$ grep Edgar telefon
Edgar Winter        x764383
Edgar Allan Poe     x728364
$
```

Beim ersten Versuch wurden keine passenden Zeilen gefunden, weil die Zeichenfolge „edgar" in der Datei nicht vorkommt.

`grep` *als Filter*

`grep` wird häufig zusammen mit anderen Befehlen verwendet, um nicht benötigte Informationen vor der Ausgabe auszufiltern. Dazu werden die von dem Befehl ausgegebenen Daten mit Hilfe des Pipe-Symbols **|** an `grep` übergeben.

Mit dem folgenden Befehl zeigen Sie alle Dateien mit der Endung `.ps` an, die im Mai erstellt wurden:

```
$ ls -l *.ps | grep Mai
```

Der erste Teil der Befehlszeile

```
ls -l *.ps
```

gibt eine Liste aller Dateien mit der Endung `.ps` im aktuellen Verzeichnis aus:

```
$ ls -l *.ps
-rw-r--r--  1 elvis       7228 Apr 22 15:07 change.ps
-rw-r--r--  1 elvis       2356 Mai 22 12:56 clock.ps
-rw-r--r--  1 elvis       1567 Jun 22 12:56 cmdtool.ps
-rw-r--r--  1 elvis      10198 Jun 22 15:07 command.ps
-rw-r--r--  1 elvis       5644 Mai 22 15:07 buttons.ps
$
```

Der zweite Teil,

```
| grep Mai
```

übergibt diese Liste an den Befehl `grep`, der die Dateien ausfiltert, in deren Beschreibung die Zeichenfolge „Mai“ vorkommt:

```
$ ls -l *.ps | grep Mai
-rw-r--r--  1 elvis       2356 Mai 22 12:56 clock.ps
-rw-r--r--  1 elvis       5644 Mai 22 15:07 buttons.ps
$
```

`grep` *mit mehrteiligen Zeichenfolgen*

Um nach Zeichenfolgen zu suchen, die aus mehr als einem Wort bestehen, setzen Sie die Zeichenfolge in einfache oder doppelte Anführungszeichen:

```
$ grep "Eduard Müller" telefon
Eduard Müller x742362
$
```

Sie können auch in mehreren Dateien nach einer Zeichenfolge suchen. Wird die Zeichenfolge gefunden, gibt `grep` den (die) Namen der Datei(en), einen Doppelpunkt und danach die gefundene Zeile aus:

```
$ grep at *
film:Walter Matthau
alaska:Alaska ist der größte Staat der USA.
computer:zeichnen sich in der Datenverarbeitung neue Trends
$
```

Ausschließende Suche

Um in einer Datei nach allen Zeilen zu suchen, in denen eine bestimmte Zeichenfolge *nicht* enthalten ist, geben Sie `grep` mit der Option `-v` ein. Im folgenden Beispiel wird in den Dateien des aktuellen Verzeichnisses nach allen Zeilen gesucht, in denen der Buchstabe `e` nicht vorkommt:

```
$ ls
alaska     film    schulung    telefon    termine
$ grep -v e *
film:Marlon Brando
telefon:Hans Schmid x834657
$
```

Weitere Suchmöglichkeiten

Sie können mit dem Befehl `grep` auch nach mehreren gleichartigen Wörtern oder Zeichenfolgen suchen, die Sie durch einen *regulären Ausdruck* spezifizieren. Reguläre Ausdrücke bestehen aus Buchstaben, Zahlen und speziellen Zeichen, die als *Metazeichen* bezeichnet und von `grep` auf eine bestimmte Weise interpretiert werden. Um eine Interpretation durch das System zu verhindern, müssen Metazeichen in der Befehlszeile in Anführungszeichen gesetzt oder mit einem Gegenschrägstrich (\) *geschützt* werden. Beispiele für Metazeichen sind:
`& ! . * $ ?` und `\`.

- Ein Zirkumflex (^) steht für den Beginn einer Zeile. Mit dem folgenden Befehl suchen Sie nach allen Zeilen in der Datei `liste`, die mit dem Buchstaben „b" beginnen:

```
$ grep '^b' liste
```

- Ein Dollarzeichen ($) steht für das Ende einer Zeile. Mit dem nächsten Befehl suchen Sie nach allen Zeilen, die mit dem Buchstaben „b" enden:

```
$ grep 'b$' liste
```

  Sie können die beiden Zeichen auch kombinieren. Mit dem folgenden Befehl suchen Sie nach allen Zeilen, in denen *nur* der Buchstabe „b" steht:

```
$ grep '^b$' liste
```

- Ein Punkt (.) steht für ein beliebiges einzelnes Zeichen. Mit dem Befehl

```
$ grep 'de.' liste
```

  suchen Sie also nach allen Zeilen, in denen eine Zeichenfolge aus „de" und einem beliebigen dritten Buchstaben enthalten ist, z. B. „der", „dem", „Katheder" oder „Ende" (Leerzeichen sind ebenfalls gültig).

- Ein Stern (*), der auf ein einzelnes Zeichen folgt, steht für eine beliebige Anzahl des vorangehenden Zeichens. Folgt der Stern auf einen regulären Ausdruck, steht er für eine beliebige Anzahl dieses Ausdrucks.

  Die Verwendung dieses Metazeichens ist mitunter nicht ganz einfach, weil der Stern auch für die Anzahl Null stehen kann. Wenn Sie beispielsweise nach allen Wörtern suchen wollen, in denen die Buchstabenfolge „qu" vorkommen, geben Sie folgendes ein:

```
$ grep 'qu*' list
```

  Soll dagegen nach allen Wörtern gesucht werden, die den Buchstaben „n" enthalten, müßten Sie folgenden Befehl eingeben:

```
$ grep 'nn*' list
```

  Um nach allen Wörter zu suchen, in denen die Buchstabenfolge „nn" vorkommt, müßten Sie den Befehl so eingeben:

```
$ grep 'nnn*' liste
```

  Am besten experimentieren Sie ein wenig mit solchen Suchmustern, um sich mit der Bedeutung der Metazeichen vertraut zu machen.

- Mit dem folgenden Befehl werden alle Zeilen in der Datei `liste` gefunden:

```
$ grep .* list
```

Nach Metazeichen suchen

Nehmen wir an, Sie wollen in einem Text nach allen Zeilen suchen, in denen ein Dollarzeichen vorkommt. Damit `grep` das Zeichen nicht als Metazeichen interpretiert, müssen Sie ihm einen Gegenschrägstrich (\) voranstellen. Auf diese Weise kann auch nach allen weiteren Metazeichen (& ! . * ? und \ selbst) gesucht werden.

Mit dem folgenden Ausdruck suchen Sie nach allen Zeilen, die mit einem Punkt beginnen:

```
$ grep ^\.
```

Auf diese Weise könnten Sie zum Beispiel in einem mit `nroff` oder `troff` formatierten Text nach Formatanweisungen suchen, da diese immer mit einem Punkt beginnen. In Tabelle 4-1 sind die in regulären Ausdrücken am häufigsten verwendeten Elemente zusammengefaßt.

Tabelle 4-1 Suchmuster-Elemente

Zeichen	**Steht für**
`^`	Beginn einer Textzeile
`$`	Ende einer Textzeile
`.`	Ein beliebiges Zeichen
`[...]`	Ein beliebiges Zeichen aus der in Klammern stehenden Liste bzw. dem in Klammern stehenden Bereich
`[^...]`	Ein beliebiges Zeichen, das in der Liste oder dem Bereich nicht vorkommt
`*`	Eine beliebige Anzahl des vorhergehenden Zeichens oder Ausdrucks
`.*`	Eine beliebige Anzahl eines beliebigen Zeichens
`\`	Hebt die spezielle Bedeutung des nachfolgenden Zeichens auf

Diese Elemente können Sie auch bei Suchläufen im Texteditor `vi` verwenden.

Anführungszeichen in der Befehlszeile

Wenn nach mehreren Wörtern gesucht werden soll, müssen diese in der Befehlszeile in Anführungszeichen gesetzt werden. Um in allen Dateien des aktuellen Verzeichnisses nach dem Ausdruck „Preise inkl. Mehrwertsteuer“ zu suchen, würden Sie also folgendes eingeben:

```
$ grep "Preise inkl. Mehrwertsteuer" *
```

Statt der doppelten können auch einfache Anführungszeichen (') verwendet werden. Damit stellen Sie gleichzeitig sicher, daß eventuell in der Zeichenfolge vorkommende Metazeichen wie normale Zeichen interpretiert werden. Das Metazeichen ! muß auch in Anführungszeichen mit einem Gegenschrägstrich entwertet werden. Um Probleme auszuschließen, sollten Sie das bei den anderen Metazeichen ebenfalls tun, wenn diese als normale Textzeichen behandelt werden sollen.

Wenn Sie folgenden Befehl eingeben, werden *alle* Zeilen in der Datei `liste` angezeigt:

```
$ grep $ liste
```

Mit dem nächsten Befehl zeigen Sie dagegen nur die Zeilen an, in denen das Zeichen „$“ vorkommt:

```
$ grep '\$' liste
```

Weitere Informationen zu `grep` finden Sie in den *man Pages(1): User Commands*.

Paßwörter und Prozesse 

SunOS bietet eine Vielzahl von Befehlen, mit denen Sie Verwaltungsaufgaben ausführen können. In diesem Kapitel erfahren Sie, wie Sie ein Paßwort setzen, eine Liste der aktiven Prozesse anzeigen, nicht mehr benötigte Prozesse abbrechen und die auf der Festplatte verfügbare Speicherkapazität abrufen.

Allgemeine Hinweise zu Paßwörtern

Damit Ihr System soweit wie möglich vor unberechtigten Zugriffen geschützt ist, müssen Sie sich bei der Anmeldung bei SunOS mit einem Paßwort ausweisen. Das Paßwort sollte mehrmals pro Jahr geändert werden, da die Möglichkeit, daß es in falsche Hände fällt, um so größer wird, je länger es in Gebrauch ist. Wenn der Verdacht besteht, daß sich ein anderer Benutzer unter Ihrem Namen und mit Ihrem Paßwort Zugang zum System verschafft hat, sollten Sie das Paßwort sofort ändern.

Bei der Auswahl des Paßworts sind folgende Hinweise zu beachten:

- Wählen Sie ein Paßwort, das Sie sich merken können, ohne es aufzuschreiben. Ein Paßwort, das möglicherweise erraten werden kann, ist immer noch besser als eines, an das Sie sich nicht mehr erinnern können, denn dann ist Ihnen der Zugang zum System versperrt.
- Das Paßwort sollte mindestens sechs Zeichen lang sein und mindestens eine Zahl enthalten.
- Verwenden Sie auf keinen Fall Ihren Namen, Ihre Initialen oder den Namen eines nahen Verwandten. Solche Paßwörter sind am leichtesten zu erraten!

- Dasselbe gilt für Namen von Haustieren oder Objekten, über die Sie am Arbeitsplatz häufig sprechen.
- Verwenden Sie nicht ausschließlich Großbuchstaben.
- Wenn Sie über mehrere Zugangsberechtigungen verfügen, sollten Sie für jede Berechtigung ein eigenes Paßwort wählen.
- Sie können zwar in einem Paßwort prinzipiell jedes beliebige Zeichen verwenden, sollten aber Zeichen wie Ctrl-C, Ctrl-Z, Ctrl-U, Ctrl-S, Esc, Tab und (mit Einschränkungen) # und @ möglichst vermeiden, da das Terminal diese Zeichen unter Umständen als Steuerzeichen interpretiert und das Paßwort dann nicht mehr korrekt erkannt wird.

Paßwort ändern

Mit dem Befehl `passwd` können Sie Ihr persönliches Paßwort ändern:

```
$ passwd
Paßwort von nina wird geändert
Old password:
New password:
Neues Paßwort noch einmal eingeben:
$
```

1. **Geben Sie auf die Eingabeaufforderung** `Old Password:` **das derzeit gültige Paßwort ein.**
   (Wenn Sie noch nicht über ein Paßwort verfügen, überspringt das System diese Eingabeaufforderung.) Das Paßwort wird aus Sicherheitsgründen bei der Eingabe nicht angezeigt.

2. **Geben Sie auf die Eingabeaufforderung** `New Password:` **das neue Paßwort ein.**
   Auch hier erscheint das Paßwort nicht auf dem Bildschirm.

3. **Geben Sie auf die Eingabeaufforderung** `Neues Paßwort noch einmal eingeben:` **das Paßwort noch einmal ein.**
   Damit soll sichergestellt werden, daß Sie sich beim ersten Mal nicht versehentlich vertippt haben. Wenn die erste und die zweite Eingabe nicht übereinstimmen, erscheint die Meldung `Bedauere`, und Sie müssen den Vorgang wiederholen. Passiert dies öfter, sollten Sie sich an Ihren Systemverwalter wenden.

Hinweis – Paßwörter, die kürzer als sechs Zeichen sind oder sich in weniger als drei Zeichen von dem alten Paßwort unterscheiden, werden nicht akzeptiert.

Paßwortalterung

Um das Paßwort möglichst geheim zu halten, kann ihm mit Hilfe von Optionen zu dem Befehl `passwd` eine Höchst- und gegebenenfalls auch eine Mindestlebensdauer zugewiesen werden. Die Zeitspanne, für die das Paßwort gültig ist, wird vom Systemverwalter festgesetzt.

Wenn Ihr Paßwort die maximale Lebensdauer erreicht, fordert das System Sie bei der nächsten Anmeldung auf, es zu ändern:

```
Your password has expired. Choose a new one.
```

Danach startet das System automatisch das Programm `passwd` und fordert Sie zur Eingabe des neuen Paßworts auf.

Wenn für Ihr Paßwort eine *Mindestlebensdauer* festgelegt wurde und Sie versuchen, es vorher zu ändern, erscheint folgende Meldung:

```
Bedauere, weniger als 14 Tage seit letzter Änderung.
```

Um den aktuellen Status Ihres Paßwortes anzuzeigen, geben Sie den Befehl `passwd` mit der Option `-d` ein:

```
$ passwd -d
username 2-14-92 14 60
```

Das erste Feld enthält Ihren Benutzernamen, dann folgen das Datum, an dem es zum letzten Mal geändert wurde und die Höchst- und Mindestlebensdauer in Tagen. Diese Informationen erscheinen nur, wenn Paßwortalterung aktiv ist.

Weitere Hinweise zu Paßwörtern und zur Paßwortalterung finden Sie in den *man Pages(1): User Commands*.

5

Prozesse und Prozeßnummern

Für jeden Befehl, den Sie an das System absetzen, wird ein eigener *Prozeß* mit einer eindeutigen Prozeßnummer (PID) erzeugt, der diesen Befehl ausführt. Mit Hilfe der Prozeßnummer kann das System jederzeit den aktuellen Status eines Prozesses ermitteln.

Prozesse anzeigen (`ps`)

Mit dem Befehl `ps` stellen Sie fest, welche Prozesse auf Ihrem System gerade ablaufen. Die Ausgabe besteht aus vier Spalten, in denen für jeden von Ihnen erzeugten Prozeß die *Prozeßnummer* (`PID`), der *Terminalname* (`TTY`), die bisher verbrauchte *CPU-Zeit* (`TIME`) und der Name des ausgeführten *Befehls* (`COMMAND`) angezeigt wird.

Wenn Sie den Befehl mit der Option `-l` eingeben, erscheinen zusätzlich weitere Informationen, darunter in der Spalte `S` der *Prozeßstatus*, der durch einen der folgenden Codes angegeben wird:

- `O` - Prozeß läuft.
- `S` - Prozeß befindet sich im Wartezustand.
- `R` - Prozeß ist ausführbar und steht in einer Warteschlange.
- `I` - Prozeß ist nicht aktiv oder wird gerade erzeugt.
- `Z` - Vater-Prozeß reagiert nicht auf Prozeßabbruch (Zombieprozeß).
- `T` - Prozeß ist angehalten und wird schrittweise ausgeführt (Trace).
- `X` -SXBRK-Status: Prozeß wartet auf Speicherzuteilung.

Der Prozeßstatus kann sich während der Ausführung von `ps` ändern, d. h. die angezeigten Werte gelten nur für den Moment, in dem der Befehl aufgerufen wurde, und können zum Zeitpunkt der Ausgabe schon wieder ungenau sein.

Weitere Optionen zu `ps` sind im *man Pages(1): User Commands* beschrieben.

Prozesse abbrechen (`kill`)

Mit dem Befehl `kill` können Sie Prozesse abbrechen, die nicht mehr benötigt werden oder die versehentlich gestartet wurden und das System zu stark belasten.

Dazu gehen Sie folgendermaßen vor:

1. **Ermitteln Sie mit** `ps` **die Nummer des Prozesses.**
2. **Geben Sie** `kill` **und die Prozeßnummer ein.**

Im folgenden Beispiel wird der Prozeß mit der Nummer `1291` abgebrochen:

```
$ ps
PID     TTY     TIME     COMMAND
1291    co      0:12     -bin/csh (csh)
3250    p0      0:00     ps
1286    p1      0:05     -bin/csh (csh)
3248    p1      0:05     vi commands
$ kill 1291
[1]  Beendet       -bin/csh/ (csh)
$
```

Schneller geht es, wenn Sie den Befehl `grep` als Filter verwenden:

```
$ ps | grep befehlsname
```

Dabei ersetzen Sie *befehlsname* durch den Namen des Befehlsprozesses, der abgebrochen werden soll.

Wenn der Prozeß auf das Abbruchsignal nicht reagiert, können Sie mit der Option `-9` ein unbedingtes Abbruchsignal senden:

```
$ kill -9 PID#
```

PID# ist die Nummer des Prozesses, der abgebrochen werden soll.

Speicherkapazität abfragen

Da Festplattenkapazität eine begrenzte Ressource ist, sollten Sie in regelmäßigen Abständen prüfen, wieviel Platz aktuell verfügbar ist.

Verfügbaren Speicherplatz anzeigen (`df -k`)

Mit dem Befehl `df -k` stellen Sie fest, wieviel Platz in den auf Ihrem System eingehängten (direkt erreichbaren) Dateisystemen noch verfügbar ist:

```
$ df -k
```

Ausgegeben werden die Gesamtkapazität, der verfügbare Speicherplatz und der bereits mit Daten belegte Speicherplatz in Prozent.

In Dateisystemen, deren Kapazität zu 90 oder mehr Prozent ausgeschöpft ist, muß „aufgeräumt" werden. Nicht mehr benötigte Dateien sollten auf eine Diskette, ein Band oder eine weniger volle Platte übertragen, mit `cp` kopiert und mit `rm` gelöscht oder gleich entfernt werden. Natürlich sollten Sie das nur mit Ihren eigenen Dateien tun und auch dann nur, wenn Sie sicher sind, daß die Dateien von anderen Benutzern nicht mehr gebraucht werden.

Verzeichnisbelegung anzeigen (`du`)

Mit dem Befehl `du` stellen Sie fest, wieviel Speicherplatz (in Blöcken zu je 512 Byte) von einem Verzeichnis und seinen Unterverzeichnissen belegt wird.

Um eine nach Größe sortierte Liste der Unterverzeichnisse eines Dateisystems anzuzeigen, wechseln Sie mit `cd` zu dem diesem Dateisystem zugeordneten Pfadnamen und geben folgenden Befehl ein:

```
$ du | sort -r -n
```

Mit Hilfe des Befehls `sort` und den Optionen `-r` (reverse) und `-n` (numeric) lassen sich große Verzeichnisse leichter lokalisieren. Mit `ls -l` können Sie dann die Größe (in Byte) und das Änderungsdatum der in solchen Verzeichnissen enthaltenen Dateien prüfen und alte oder sehr große Dateien gegebenenfalls auslagern.

Textbearbeitung mit dem Editor `vi` *6* 

`vi` (*vi*sual display editor) ist der unter SunOS standardmäßig verwendete Texteditor. Da er zeichenorientiert und nicht fensterorientiert arbeitet, kann er auf allen Terminaltypen eingesetzt werden und unterstützt eine Vielzahl von Dateiformaten.

Sie können mit `vi` zwar Text eingeben und bearbeiten, aber nicht auf dem Bildschirm formatieren. `vi` ist deshalb kein Ersatz für ein echtes Textverarbeitungsprogramm. Um in `vi` eingegebenen Text für den Ausdruck zu formatieren, benötigen Sie ein Programm wie `nroff`, `troff` oder `ditroff`, das die in den Text eingefügten Formatierbefehle liest und in entsprechende Formate umsetzt.

Von dieser Einschränkung einmal abgesehen, ist `vi` jedoch ein außerordentlich mächtiger Editor, und neue Benutzer fühlen sich angesichts der Fülle teilweise überlappender Befehle und Funktionen zunächst oft überfordert. Dieses Kapitel soll daher nur einen Überblick über die wichtigsten `vi`-Befehle geben, mit denen Sie sich dann im Laufe der Zeit vertraut machen können.

Von `vi` gibt es auch eine Art Demoversion namens `view`. Wenn Sie eine Datei mit `view` öffnen, stehen zwar alle `vi`-Befehle zur Verfügung, die Datei kann jedoch nicht gespeichert werden. Auf diese Weise lassen sich versehentliche Änderungen am Dateiinhalt ausschließen.



vi starten

In den folgenden Abschnitten erfahren Sie, wie Sie `vi` aufrufen, Text in eine Datei eingeben, die Datei speichern und `vi` beenden. Dabei erstellen Sie eine Übungsdatei, mit der Sie im weiteren Verlauf des Kapitels arbeiten.

Dateien erstellen

Mit dem folgenden Befehl starten Sie `vi` und weisen das Programm gleichzeitig an, die Datei `test` zu öffnen:

```
$ vi test
```

Existiert die Datei bereits, wird sie geladen. Ist eine Datei dieses Namens noch nicht vorhanden, erstellt der Editor sie. In den nachfolgenden Beispielen wird vorausgesetzt, daß `test` eine neue Datei ist.

Nach dem Aufruf von `vi` erscheint folgender Bildschirm:

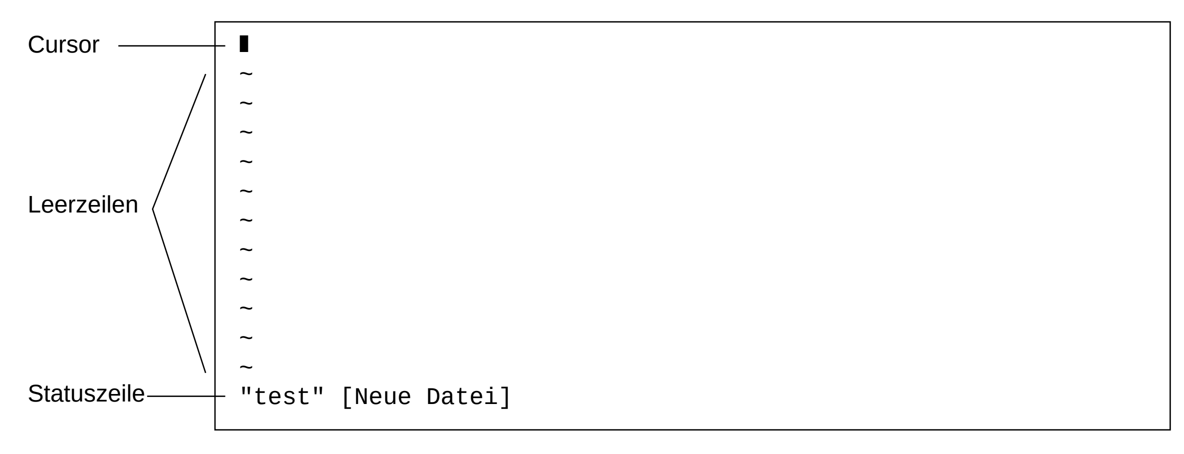


Abbildung 6-1 Der Editor-Bildschirm

Der Cursor wird in der oberen linken Ecke des Bildschirms angezeigt. Jede Tilde (~) steht für eine Leerzeile.

Sie können `vi` auch ohne Angabe eines Dateinamens starten und der Datei erst beim Verlassen des Editors einen Namen zuweisen.

Statuszeile

In der *Statuszeile* am unteren Bildschirmrand werden der Dateiname und die Anzahl der Zeilen und Zeichen in der Datei angezeigt. Wenn Sie eine neue Datei erstellen, erscheint in der Statuszeile die Meldung `Neue Datei`.

Betriebsmodus

`vi` unterstützt zwei Betriebsmodi: den Eingabemodus und den Befehlsmodus. Im *Eingabemodus* geben Sie Text in eine Datei ein, während der *Befehlsmodus* für die Eingabe von Befehlen vorgesehen ist, mit denen spezielle Editorfunktionen aufgerufen werden.

Da `vi` den aktuellen Modus nicht anzeigt, wissen neue Benutzer häufig nicht, in welchem Modus sie sich gerade befinden, was dann leicht zu Verwirrung führt. `vi`-Probleme lassen sich jedoch weitgehend vermeiden, wenn Sie einige grundlegende Hinweise von Anfang an beachten.

Wenn Sie eine Datei das erste Mal öffnen, befindet sich `vi` im Befehlsmodus. Bevor Sie Text in die Datei eingeben können, müssen Sie also in den Eingabemodus wechseln. Dafür gibt es mehrere Befehle, z. B. `i` (insert), mit dem Sie Text *an* der Cursorposition einfügen, oder `a` (append), mit dem Sie Text *nach* der Cursorposition einfügen. (Diese und weitere Eingabebefehle werden später noch ausführlich behandelt.)

Mit der Taste „Esc“ kehren Sie in den Befehlsmodus zurück. Wenn Sie nicht sicher sind, in welchem Modus Sie sich befinden, können Sie mit „Esc“ also zunächst in den Befehlsmodus wechseln und dann gegebenenfalls wieder zum Eingabemodus umschalten. Befindet sich `vi` bereits im Befehlsmodus, wenn Sie „Esc“ drücken, blinkt der Bildschirm, und es wird ein akustisches Signal ausgegeben, das Sie aber ignorieren können.

Eingabemodus

Um mit der Texteingabe zu beginnen, geben Sie den Befehl `i` ein. Damit wechseln Sie in den Eingabemodus.

Geben Sie nun einige kurze Textzeilen ein, und beenden Sie jede Zeile mit „Return". Die eingegebenen Zeichen erscheinen links vom Cursor, bereits vorhandene Zeichen werden dabei nach rechts verschoben. Tippfehler korrigieren Sie zunächst durch Löschen mit der Rücktaste (Back Space) und Neueingabe der Zeichen vor Beendigung einer Zeile mit „Return".

Wenn Sie mit der Eingabe fertig sind, wechseln Sie mit „Esc" zurück in den Befehlsmodus. Der Cursor springt dabei auf das zuletzt eingegebene Zeichen zurück.

Reagiert `vi` nicht wie erwartet, haben Sie möglicherweise versehentlich auf Großbuchstaben-Eingabe (Caps) umgeschaltet. Auf manchen Systemen wird dieser Modus mit der Taste „F1" aktiviert, die sich normalerweise direkt neben der Taste „Esc" befindet und daher oft anstelle von „Esc" gedrückt wird.

Hinweis – Wenn auf dem Bildschirm „verirrte" Systemmeldungen oder andere unerwünschte Zeichen auftauchen, wechseln Sie in den Befehlsmodus und drücken „Ctrl-L". Damit weisen Sie `vi` an, den Bildschirminhalt neu zu zeichnen. Dieser Befehl ähnelt dem OpenWindows-Dienstprogramm „Aktualisieren".

Befehlsmodus

Nach dem Öffnen einer Datei befindet sich `vi` im Befehlsmodus. In diesem Modus können Sie Befehle eingeben, mit denen bestimmte Funktionen ausgelöst werden. Die meisten Editorbefehle bestehen aus einem oder zwei Buchstaben und mitunter auch einer Zahl. Groß- und Kleinbuchstaben haben bei `vi`-Befehlen meist unterschiedliche Bedeutung. Mit dem Befehl `a` fügen Sie beispielsweise Zeichen rechts vom Cursor ein, während Sie mit dem Befehl `A` Zeichen am Ende der Zeile einfügen.

Die meisten Befehle werden automatisch ausgeführt. „Return" müssen Sie nur bei Befehlen drücken, die mit einem Doppelpunkt beginnen. Da diese Befehle bei der Eingabe in der Statuszeile angezeigt werden, spricht man mitunter auch vom *Statuszeilenmodus* als drittem Betriebsmodus. In diesem Handbuch wird jedoch nur zwischen Eingabe- und Befehlsmodus unterschieden.

Mit einem Doppelpunkt beginnende Befehle sind eigentlich *ex*-Befehle. vi und ex sind jedoch keine separaten Programme, sondern lediglich zwei verschiedene Schnittstellen zu dem gleichen Textbearbeitungsprogramm. vi ist eine bildschirmorientierte Schnittstelle, während ex zeilenorientiert arbeitet. Wenn Sie im Befehlsmodus einen Doppelpunkt eingaben, schalten Sie automatisch auf die Schnittstelle ex um und können so auf den gesamten Funktionsumfang von ex zugreifen, ohne vi verlassen zu müssen. Weitere Hinweise zu diesem Thema finden Sie im Abschnitt „ex-Befehle".

Arbeitssitzungen beenden

Bei der Bearbeitung einer Datei mit vi werden die Änderungen nicht direkt in die Datei geschrieben, sondern in eine von vi erzeugte Kopie, die sich in einem temporären Speicherbereich, dem *Dateipuffer*, befindet. Wenn Sie den Inhalt des Puffers speichern, wird er in die auf der Platte gespeicherte Datei übernommen.

Dieses Verfahren hat Vor- und Nachteile. Einerseits können Sie die Bearbeitung einer Datei jederzeit abbrechen, ohne die Kopie auf der Festplatte zu ändern, andererseits geht der ungesicherte Inhalt des Puffers bei einem Systemabsturz verloren. (Von solchen unvorhergesehenen Unterbrechungen sind vor allem Benutzer entfernter Terminals betroffen, die über Telefonleitungen mit dem Server verbunden sind.)

Am besten schützen Sie sich vor Datenverlusten, wenn Sie Ihre Arbeit regelmäßig, zumindest aber jedesmal nach einer größeren Änderung sichern.



Vorsicht – Sie können eine Datei im Prinzip in mehreren, gleichzeitig ablaufenden vi-Sitzungen bearbeiten. Weil Sie dabei aber häufig nicht mehr eindeutig nachvollziehen können, welche Änderungen bereits in die Datei geschrieben bzw. von anderen Sitzungen überschrieben wurden, ist dieses Verfahren nicht zu empfehlen.

Änderungen speichern und vi beenden

vi verfügt über eine Vielzahl von Befehlen zum Speichern des Pufferinhalts und zum Verlassen des Editors, die nur geringfügig voneinander abweichen. Grundsätzlich unterscheidet man zwischen Speichern, Speichern und Beenden und Beenden ohne Speichern.

Speichern

Um den Inhalt des Puffers in die Datei auf der Platte zu schreiben, geben Sie folgenden Befehl ein:

```
:w
```

Da der Befehl mit einem Doppelpunkt beginnt, müssen Sie zum Abschluß „Return“ drücken.

Speichern und Beenden

Mit folgendem Befehl speichern Sie den Inhalt des Puffers und beenden `vi`:

```
:wq
```

Sie können statt dessen auch `ZZ` eingeben. In diesem Fall schließen Sie die Eingabe nicht mit „Return“ ab.

Beenden ohne Speichern

Wenn Sie die Datei nicht geändert haben und den Editor verlassen wollen, geben Sie folgenden Befehl ein:

```
:q
```

Auch hier müssen Sie die Eingabe mit „Return“ abschließen. Wenn Sie Änderungen vorgenommen haben, erscheint die Meldung `Kein Schreibvorgang seit letzter Änderung (:quit! setzt außer Kraft)`, und `vi` wird nicht beendet.

Sollen vorhandene Änderungen nicht übernommen werden, geben Sie folgenden Befehl ein und drücken „Return“:

```
:q!
```

Dateien drucken

Um eine gespeicherte Datei zu drucken, verlassen Sie `vi` und geben folgenden Befehl ein:

```
$ lp dateiname
```

dateiname steht für den Namen der Datei, die gedruckt werden soll. Die Datei wird auf dem Standarddrucker ausgegeben, und zwar unformatiert, also genau so, wie sie auf dem Bildschirm erscheint. Weitere Informationen zu Druckbefehlen finden Sie in Kapitel 8, „Drucken".

`vi`-*Befehle*

In den folgenden Abschnitten werden `vi`-Befehle für verschiedene Aufgabenstellungen behandelt:

- Navigationsbefehle
- Text einfügen
- Text ändern und ersetzen
- Änderungen rückgängig machen
- Text löschen
- Text kopieren und verschieben
- Befehle wiederholen

Navigationsbefehle

Navigationsbefehle sind Befehle, mit denen Sie den Cursor durch den Text bewegen oder unterschiedliche Ausschnitte einer Datei auf dem Bildschirm anzeigen. Da diese Befehle für die Textbearbeitung besonders wichtig sind, sollten Sie sie beim Lesen der folgenden Abschnitte in Ihrer Beispieldatei ausprobieren.

Cursor bewegen

Nach dem Start von `vi` befindet sich der Cursor in der oberen linken Ecke des Bildschirms. Im Befehlsmodus bewegen Sie den Cursor mit den Pfeiltasten, der Leertaste und einigen anderen Tasten sowie durch Eingabe von Befehlen.

Hinweis – `vi`-Befehle sind kontextabhängig. Wenn Sie den „gleichen“ Befehl in Großbuchstaben und Kleinbuchstaben eingeben, kann es also zu sehr unterschiedlichen Ergebnissen kommen.

Um ein Zeichen (Pfeiltasten)

Wenn auf Ihrer Tastatur Pfeiltasten vorhanden sind, können Sie den Cursor damit beliebig nach oben, unten, rechts und links durch den vorhandenen Text (und die angezeigten Leerzeilen) bewegen.

Wenn Sie auf einem entfernten Terminal mit `vi` arbeiten, funktioniert diese Methode mitunter nicht, da manche Terminalemulatoren die Pfeiltasten nicht unterstützen. In diesem Fall können Sie ersatzweise folgende Tasten verwenden:

- Mit `h` bewegen Sie den Cursor um ein Zeichen nach links.
- Mit `l` bewegen Sie den Cursor um ein Zeichen nach rechts.
- Mit `j` bewegen Sie den Cursor um ein Zeichen nach unten.
- Mit `k` bewegen Sie den Cursor um ein Zeichen nach oben

Um ein Wort

Mit `w` bewegen Sie den Cursor zum nächsten Wort oder Satzzeichen.

Mit `b` bewegen Sie den Cursor zum vorhergehenden Wort oder Satzzeichen.

Mit `W` oder `B` setzen Sie den Cursor auf das Leerzeichen vor dem nächsten bzw. vorherigen Wort. Satzzeichen werden dabei übersprungen.

Mit `e` bewegen Sie den Cursor auf das letzte Zeichen des aktuellen Worts.

An den Zeilenanfang oder das Zeilenende

Mit `^` bewegen Sie den Cursor auf das erste Zeichen der aktuellen Zeile.

Mit `$` bewegen Sie den Cursor auf das letzte Zeichen der aktuellen Zeile.

Um eine Zeile nach unten

Mit „Return“ bewegen Sie den Cursor auf den Anfang der nächsten Zeile.

Nach links

Mit der Rücktaste bewegen Sie den Cursor um ein Zeichen nach links.

Nach rechts

Mit der Leertaste bewegen Sie den Cursor um ein Zeichen nach rechts.

Nach oben

Mit `H` bewegen Sie den Cursor an den Anfang des Bildschirms.

In die Mitte

Mit `M` bewegen Sie den Cursor in die Mitte des Bildschirms.

Nach unten

Mit `L` bewegen Sie den Cursor an das Ende des Bildschirms.

Bildschirminhalt verschieben

`vi` zeigt immer eine Bildschirmseite aus der Textdatei an. Wenn Sie sich am oberen oder unteren Bildschirmrand befinden und den Cursor weiter nach oben oder unten bewegen, wird der Text in die entsprechende Richtung bewegt, so daß vorher nicht sichtbare Zeilen erscheinen. Dieses Verfahren ist bei langen Dateien jedoch sehr umständlich.

Sie können den Bildschirminhalt auch gleich um eine ganze oder halbe Bildschirmseite nach oben oder unten verschieben. (Wenn Sie das in Ihrer Beispieldatei ausprobieren möchten, müssen Sie weiteren Text eingeben.)

Auf einem schnellen System bestehen zwischen dem *zeilenweisen* und *bildschirmweisen* Verschieben oft keine großen Geschwindigkeitsunterschiede. Auf einem entfernten Terminal oder einer durch andere Prozesse bereits stark belasteten Workstation nimmt die Geschwindigkeit beim zeilenweisen Verschieben dagegen stark ab.

Eine Bildschirmseite vorwärts

Mit „Ctrl-F“ zeigen Sie die nächste Bildschirmseite an. Der Cursor springt dabei auf das erste Zeichen der neuen Seite.

Eine halbe Bildschirmseite nach unten

Mit „Ctrl-D" verschieben Sie den Bildschirminhalt um eine halbe Seite nach unten.

Eine Bildschirmseite zurück

Mit „Ctrl-B" zeigen Sie die vorhergehende Bildschirmseite an.

Eine halbe Bildschirmseite nach oben

Mit „Ctrl-U" verschieben Sie den Bildschirminhalt um eine halbe Seite nach oben.

Text eingeben

`vi` verfügt über eine ganze Reihe von Befehlen für die Texteingabe, von denen hier nur die wichtigsten vorgestellt werden. Mit jedem dieser Befehle wechseln Sie gleichzeitig in den Eingabemodus. Wenn Sie sich bereits im Eingabemodus befinden, müssen Sie zuerst mit „Esc" in den Befehlsmodus zurückkehren.

Text anhängen

Mit dem Befehl `a` legen Sie fest, daß der danach eingegebene Text *rechts* vom Cursor erscheinen soll. Bewegen Sie den Cursor jetzt auf eine Leerzeile, geben Sie `a` und danach den gewünschten Text ein, und drücken Sie nach der Eingabe „Esc".

Soll der eingegebene Text *am Ende* der aktuellen Zeile erscheinen, geben Sie statt dessen `A` ein. Bewegen Sie den Cursor auf eine Textzeile, geben Sie `A` und den gewünschten Text ein und drücken Sie zum Schluß „Esc".

Text einfügen

Mit dem Befehl `i` legen Sie fest, daß der neue Text links vom Cursor eingesetzt werden soll. Wenn Sie statt dessen `I` eingeben, wird der Text am Anfang der aktuellen Zeile eingefügt. Probieren Sie beide Befehle aus, und wechseln Sie danach mit „Esc" in den Befehlsmodus zurück.

Leerzeilen einfügen

Mit den folgenden Befehlen fügen Sie über oder unter der aktuellen Cursorposition neue Zeilen ein.

Mit `o` öffnen Sie eine Zeile *unter* der aktuellen Zeile. Geben Sie jetzt `o` und ein oder mehrere Textzeilen ein, und drücken Sie zum Schluß wieder „Esc".

Mit `O` öffnen Sie eine Zeile *über* der aktuellen Zeile.

Text ändern

Wenn Sie Text ändern, ersetzen Sie einen Textblock durch einen anderen. `vi` bietet dafür mehrere Möglichkeiten.

Wörter ersetzen

Um ein Wort durch ein anderes zu ersetzen, setzen Sie den Cursor im Befehlsmodus an den Anfang des Worts und geben `cw` und danach das neue Wort ein. Danach wechseln Sie mit „Esc" zurück in den Befehlsmodus.

Um ein Wort teilweise zu ersetzen, bewegen Sie den Cursor auf das erste Zeichen des zu löschenden Teils, geben `cw` und die neue Zeichenfolge ein und drücken zum Schluß „Esc".

Zeilen ersetzen

Um eine Zeile zu ersetzen, bewegen Sie den Cursor auf ein beliebiges Zeichen in der Zeile und geben `cc` ein. Die Zeile wird geleert, d. h. es bleibt eine Leerzeile zurück, in die Sie neuen Text eingeben können. Drücken Sie zum Abschluß „Esc".

Zeilen teilweise ersetzen

Um einen Teil einer Zeile zu ersetzen, bewegen Sie den Cursor auf das erste Zeichen des zu löschenden Teils, geben `C` und den neuen Text ein und drücken zum Schluß „Esc". Damit werden alle Zeichen von der Cursorposition bis zum Ende der Zeile gelöscht und durch den neu eingegebenen Text ersetzt.

Zeichen ersetzen

Um ein Zeichen durch eines oder mehrere andere Zeichen zu ersetzen, bewegen Sie den Cursor auf das Zeichen, geben `s` und die neuen Zeichen ein und drücken „Esc", um in den Befehlsmodus zurückzukehren.

Einzelne Zeichen ersetzen

Wenn Sie ein Zeichen durch ein einzelnes anderes Zeichen ersetzen wollen, bewegen Sie den Cursor auf das zu löschende Zeichen und geben `r` und das neue Zeichen ein. Bei diesem Befehl brauchen Sie „Esc" nicht zu drücken, da `vi` automatisch in den Befehlsmodus zurückkehrt.

Zeichen vertauschen

„Verdreher" lassen sich mit `vi` recht einfach korrigieren. Nehmen wir an, Sie haben "`udn`" statt "`und`" eingegeben. Um den Fehler zu korrigieren, bewegen Sie den Cursor auf das erste Zeichen, das vertauscht werden soll (also das `d`) und geben `xp` ein. Das `d` und das `h` werden vertauscht, und `vi` wechselt automatisch zurück in den Befehlsmodus.

Zeilen teilen oder zusammenführen

Um eine Zeile zu teilen, ohne den Text zu verändern, setzen Sie den Cursor auf ein Leerzeichen in der Zeile, geben `r` ein und drücken „Return". Steht der Cursor dabei auf einem Zeichen, wird es durch das Zeichen „Return" (also eine Zeilenschaltung) ersetzt.

Um zwei Zeilen zusammenzuführen, setzen Sie den Cursor auf die obere Zeile und geben `J` ein. `vi` bleibt dabei im Befehlsmodus, so daß Sie „Esc" nicht zu drücken brauchen.

Änderungen aufheben

Wenn Sie Text versehentlich ändern oder eine Änderung nicht das gewünschte Ergebnis hat, können Sie mit einem der folgenden Befehle den ursprünglichen Zustand wiederherstellen.

Letzten Befehl rückgängig machen

Mit dem Befehl `u` machen Sie den zuletzt eingegebenen Befehl rückgängig. (`vi` bleibt dabei im Befehlsmodus.) Wenn Sie `u` ein zweites Mal drücken, wird `u` selbst rückgängig gemacht, d. h. die Änderung wird wiederhergestellt.

Zeilenänderungen rückgängig machen

Mit dem Befehl `U` nehmen Sie alle Änderungen zurück, die Sie in der aktuellen Zeile gemacht haben, seitdem sich der Cursor darin befindet. Auch bei diesem Befehl brauchen Sie nach der Eingabe nicht „Esc“ zu drücken.

Text löschen

Mit den folgenden Befehlen löschen Sie Zeichen, Wörter oder ganze Zeilen. `vi` bleibt dabei im Befehlsmodus, so daß Sie vor Eingabe neuer Zeichen zuerst in den Eingabemodus wechseln müssen.

Einzelne Zeichen löschen

Mit dem Befehl `x` löschen Sie das Zeichen an der Cursorposition.

Wenn Sie ein Zeichen innerhalb einer Zeile löschen, werden die rechts davon stehenden Zeichen um eine Stelle nach links verschoben, so daß keine Lücke zurückbleibt. Sie können mit `x` auch Leerzeichen löschen.

Um das Zeichen links vom Cursor zu löschen, geben Sie `X` ein.

Wörter ganz oder teilweise löschen

Um ein Wort zu löschen, setzen Sie den Cursor auf das erste Zeichen des Worts und geben `dw` ein. Soll nur ein Teil des Worts gelöscht werden, setzen Sie den Cursor auf das erste zu löschende Zeichen.

Zeilen löschen

Um eine Zeile zu löschen, setzen Sie den Cursor auf ein beliebiges Zeichen in der Zeile und geben `dd` ein. Damit löschen Sie nicht nur den Text in der Zeile, sondern auch die Zeile selbst.

Zeilen teilweise löschen

Wenn Sie nur einen Teil einer Zeile löschen wollen, setzen Sie den Cursor auf das erste zu löschende Zeichen und geben `D` ein. Damit löschen Sie alle Zeichen von der Cursorposition bis zum Zeilenende.

Um alle Zeichen von der Cursorposition bis zum Zeilenanfang zu löschen, geben Sie statt dessen `d0` (`d`-Null) ein.

Bis zum Dateiende löschen

Um den gesamten Text von der aktuellen Zeile bis zum Dateiende zu löschen, geben Sie `dG` ein.Die aktuelle Zeile wird dabei ebenfalls gelöscht.

Bis zum Dateianfang löschen

Um den gesamten Text von der aktuellen Zeile bis zum Dateianfang zu löschen, geben Sie `d1G` ein. Die aktuelle Zeile wird dabei ebenfalls gelöscht.

Text kopieren und übertragen

In vielen Textverarbeitungsprogrammen können Sie Text kopieren oder ausschneiden und an einer anderen Stelle wieder einsetzen. `vi` bietet die gleichen Funktionen, die im Englischen *yank and put* (Kopieren und Einsetzen) und *delete and put* (Ausschneiden und Einsetzen) heißen.

Zeilen kopieren

Beim Kopieren von Zeilen verwenden Sie zwei Befehle: `yy` oder `Y` (diese beiden Befehle haben die gleiche Funktion) und `p` (unten einsetzen) oder `P` (oben einsetzen).

Um die Zeile zu kopieren, setzen Sie den Cursor auf ein beliebiges Zeichen in der Zeile und geben `yy` ein. Danach bewegen Sie den Cursor auf die Zeile, unter der die kopierte Zeile eingefügt werden soll, und geben `p` ein.

Soll die kopierte Zeile *über* der aktuellen Zeile eingesetzt werden, geben Sie `P` ein.

Der Befehl `yy` kann auch auf mehrere Zeilen angewendet werden: Mit `11yy` kopieren Sie beispielsweise die aktuelle Zeile und die folgenden zehn Zeilen. Dabei erscheint in der Statuszeile die Meldung `11 Zeilen kopiert`.

Die Befehle `P` und `p` können auch zusammen mit den zuvor beschriebenen Löschbefehlen verwendet werden, um gelöschten Text über oder unter der Cursorposition wieder einzusetzen.



Vorsicht – Da gelöschte oder kopierte Textblöcke in einen Puffer übertragen werden, dürfen Sie vor dem Einsetzen eines Blocks nur Befehle verwenden, mit denen der Cursor bewegt wird. Wenn Sie einen anderen Textblock löschen oder kopieren, wird der Inhalt des Puffers überschrieben.

Zeilen bewegen

Beim Bewegen von Zeilen verwenden Sie ebenfalls zwei Befehle: `dd` und `p` oder `P`.

Um eine Zeile zu bewegen, setzen Sie den Cursor auf die Zeile und geben `dd` ein. Damit löschen Sie die Zeile und übertragen sie in den Puffer. (Sie können auch mehrere Zeilen löschen. Mit `5dd` löschen Sie beispielsweise die aktuelle Zeile und die darunterstehenden vier Zeilen.)

Danach bewegen Sie den Cursor auf die Zeile, unter der die gelöschte Zeile eingefügt werden soll, und geben `p` ein.

Soll die gelöschte Zeile *über* der aktuellen Zeile eingesetzt werden, geben Sie `P` ein.

Benannte Puffer

Der Inhalt des Standardpuffers wird, wie bereits erwähnt, jedesmal überschrieben, wenn Sie einen neuen Textblock löschen oder kopieren. Wenn Sie eine oder mehrere Zeilen an verschiedenen Stellen einer Datei einsetzen wollen,

können Sie sie in einen benannten Puffer löschen bzw. kopieren. Der Inhalt eines benannten Puffers wird erst überschrieben, wenn Sie eine andere Zeilengruppe explizit darin ablegen.

Um eine oder mehrere Zeilen in einem benannten Puffer zu speichern, stellen Sie vor den Lösch- oder Kopierbefehl ein doppeltes Anführungszeichen (`"`) und den Namen des Puffers. Mit dem Befehl `"a4yy` kopieren Sie beispielsweise die aktuelle Zeile und die folgenden drei Zeilen in einen Puffer namens „a“. Mit dem Befehl `"b12dd` löschen Sie 12 Zeilen in einen Puffer namens „b“. Sie können auch mehrere Puffer verwenden.

Um Text aus einem Puffer in die Datei einzusetzen, stellen Sie vor den Befehl `p` oder `P` ein doppeltes Anführungszeichen und den Namen des Puffers (z. B. `"bP`).

Wenn Sie die aktuelle Arbeitssitzung beenden, wird der Inhalt aller benannten Puffer gelöscht.

Befehle wiederholen

Viele `vi`-Befehle können zusammen mit einem *Zähler* eingegeben werden, d. h. mit einer Zahl, die vor dem Befehl steht und angibt, wie oft der Befehl wiederholt werden soll.

Die meisten der bisher behandelten Befehle unterstützen Zähler. Mit `3dd` weisen Sie `vi` beispielsweise an, den Befehl `dd` dreimal auszuführen und so die aktuelle Zeile und die beiden darauffolgenden Zeilen zu löschen. Mit `2dw` löschen Sie zwei Wörter, mit `4x` vier Zeichen usw. Zähler können auch bei der Cursorsteuerung verwendet werden. Mit `2`„Ctrl-F“ wechseln Sie zum Beispiel zur übernächsten Bildschirmseite. In den Beschreibungen der weiteren `vi`-Befehle werden Sie andere Möglichkeiten für die Verwendung von Zählern kennenlernen. Im Abschnitt „Befehlsübersicht“ am Ende dieses Kapitels ist zu Befehlen, die Zähler unterstützen, ein entsprechendes Beispiel angegeben.

Durch Eingabe eines Punkts (`.`) wiederholen Sie den letzten Textbearbeitungsbefehl. Wenn Sie beispielsweise gerade eine Zeile mit `dd` gelöscht haben und eine weitere Zeile löschen wollen, setzen Sie den Cursor in diese Zeile und geben einen Punkt ein.

ex-Befehle

Wenn Sie mit größeren Textblöcken arbeiten, sind ex-Befehle praktischer als die zuvor beschriebenen vi-Befehle. Anstatt die zu löschenden oder zu kopierenden Zeilen zu zählen und dann einen Einfügepunkt zu setzen, geben Sie einfach einen Zeilenbereich an und legen fest, nach welcher Zeilennummer diese Zeilen eingesetzt werden sollen.

Zeilennummern anzeigen

Um die Bildschirmzeilen zu numerieren, geben Sie type `:set nu` ein und drücken „Return“.

Die Zeilennummern werden am linken Bildschirmrand angezeigt. Wenn Sie die Datei drucken, werden die Nummern nicht ausgegeben.

```
1 Wir träumen von Reisen durch das Weltall -
2 Ist denn das Weltall nicht in uns?
3 Die Tiefen unseres Geistes kennen wir nicht -
4 Nach innen der geheimnisvolle Weg.
5 In uns, oder nirgends ist die Ewigkeit mit ihren Welten -
6 die Vergangenheit und Zukunft.
7
8 Novalis
~
~
~
~
~
~
:set nu
```

Um die Ausgabe der Zeilennummern zu deaktivieren, geben Sie `:set nonu` ein und drücken „Return“.

Zeilen kopieren

Der Kopierbefehl von ex wird in folgendem Format eingegeben:

```
:zeile#,zeile# co zeile#
```

Die beiden ersten, durch ein Komma getrennten Zeilennummern geben den zu kopierenden Bereich an. Die dritte Nummer bezeichnet die Zeile, *unter* der die kopierten Zeilen eingefügt werden sollen.

Um die Zeilen 1 bis 5 in der Datei `test` zu kopieren und unter der Zeile 12 einzusetzen, würden Sie also folgendes eingeben:

```
:1,5 co 12
```

Schließen Sie die Eingabe mit „Return“ ab.

Bei der Angabe des Zeilenbereichs können Sie zwei Abkürzungen verwenden:

- Der Punkt (`.`) steht für die aktuelle Zeile.
- Das Dollarzeichen ($) steht für das Dateiende.

Mit dem folgenden Befehl kopieren Sie den Bereich von der aktuellen Zeile bis Zeile 5 und setzen die kopierten Zeilen nach Zeile 12 ein:

```
:.,5 co 12
```

Mit dem folgenden Befehl kopieren Sie alle Zeilen von Zeile 6 bis zum Dateiende und setzen die kopierten Zeilen nach Zeile 2 ein:

```
:6,$ co 2
```

Zeilen bewegen

Der Befehl zum Bewegen von Zeilen ist dem oben beschriebenen Kopierbefehl sehr ähnlich:

```
:zeile#,zeile# m zeile#
```

Zeilenbereich und Einfügepunkt werden ebenso angegeben wie beim Kopieren von Zeilen. Die Abkürzungen `.` und `$` sind ebenfalls zulässig. Der Bereich wird hier jedoch nicht kopiert, sondern ausgeschnitten und an einer anderen Stelle wieder eingesetzt.

Um die Zeilen 1 bis 5 zu löschen und nach Zeile 12 wieder einzusetzen, geben Sie folgenden Befehl ein und drücken „Return“:

```
:1,5 m 12
```

Zeilen löschen

Mit folgendem Befehl löschen Sie eine Gruppe von Zeilen:

```
:zeile#,zeile# d
```

Um die Zeilen 1 bis 5 zu löschen, geben Sie also folgendes ein:

```
:1,5 d
```

Text suchen und ersetzen

`vi` bietet mehrere Befehle, mit denen Sie innerhalb einer Datei nach Zeichenfolgen suchen und diese durch andere Zeichenfolgen ersetzen können.

Zeichenfolgen suchen

Eine *Zeichenfolge* ist eine beliebige Kombination aus Buchstaben, Zahlen, Satzzeichen und Sonderzeichen wie Tabs, Zeilenschaltungen und Leerzeichen. Zeichenfolgen können aus einem Teil eines Worts, einem vollständigen Wort oder mehreren Wörtern bestehen.

Um ab der aktuellen Cursorposition in Richtung Dateiende nach einer Zeichenfolge zu suchen, geben Sie einen Schrägstrich (`/`) und den Suchtext ein und schließen die Eingabe mit „Return“ ab. Der Cursor springt daraufhin auf die nächste Stelle, an der die Zeichenfolge vorkommt.

Wenn in Richtung Dateianfang gesucht werden soll, geben Sie statt `/` das Zeichen `?` ein.

Danach können Sie den Suchvorgang mit `n` und `N` beliebig oft wiederholen, wobei `n` immer in Standardrichtung und `N` in umgekehrter Richtung sucht.

Normalerweise wird Groß- und Kleinschreibung beachtet, d. h. wenn Sie nach „modus“ suchen, wird „Modus“ nicht gefunden. Soll die Schreibweise ignoriert werden, geben Sie `:set ic` ein. Diese Einstellung gilt dann solange, bis Sie sie mit `:set noic` deaktivieren.

Ist die gesuchte Zeichenfolge nicht vorhanden, wird in der Statuszeile die Meldung `Muster nicht gefunden` angezeigt.

Einige Sonderzeichen (`/ & ! . ^ * $` \ ?) haben bei Suchläufen spezielle Bedeutung und müssen daher durch einen vorangestellten Gegenschrägstrich entwertet werden, wenn sie als normale Textzeichen interpretiert werden sollen. Um beispielsweise nach der Zeichenfolge „wieso?“ zu suchen, geben Sie `/wieso\?` ein und drücken „Return“. Um einen Gegenschrägstrich zu entwerten, geben Sie \\ ein.

Die Verwendung dieser Sonderzeichen als Befehle bei einem Suchlauf wird im folgendem Abschnitt beschrieben.

Suchoptionen

Um genauer zu suchen, können Sie in der Zeichenfolge bestimmte Suchkriterien durch Sonderzeichen angeben:

- Zeilenanfang
- Zeilenende
- Wortanfang
- Wortende
- Metazeichen

Der Zirkumflex (^) steht für den Anfang einer Zeile. Mit folgendem Befehl suchen Sie nach der nächsten Zeile, die mit der Zeichenfolge „Test“ beginnt:

```
/^Test
```

Das Dollarzeichen ($) steht für das Ende einer Zeile. Mit folgendem Befehl suchen Sie nach der nächsten Zeile, die mit „test.“ endet (da der Punkt auch als Suchkriterium verwendet werden kann, muß er entwertet werden):

```
/test\.$
```

Die Zeichenkombinationen \< und \> stehen für den Anfang bzw. das Ende eines Worts und werden an den Anfang und das Ende des Suchtexts gestellt. Auf diese Weise können Sie den Editor anweisen, nach vollständigen Wörtern zu suchen. Bei Eingabe des folgenden Befehls wird also nur das Wort „Test" selbst gefunden, andere Wörter, die mit „Test" beginnen, dagegen nicht:

```
/\<Test\>
```

Der Punkt (`.`) steht für ein beliebiges Zeichen. Mit dem folgenden Befehl wird sowohl nach „Test" wie nach „Text" gesucht:

```
/Te.t
```

Da Sie hier nach einer Zeichenfolge und nicht nach einem Wort suchen, werden auch Wörter wie „Testlauf" oder „Textverarbeitung" gefunden.

Es gibt noch eine weitere, vielseitigere Möglichkeit, nach verschiedenen Schreibweisen einer Zeichenfolge zu suchen. Dabei setzen Sie die alternativ zu verwendenden Zeichen in eckige Klammern. Wenn Sie `/[mh]`*zeichenfolge* eingeben, kann die gesuchte Zeichenfolge also mit „m" oder „h" beginnen. Geben Sie statt dessen `/[d-m]`*zeichenfolge* ein, kann die Zeichenfolge mit einem beliebigen Buchstaben von „d" bis „m" beginnen.

Der Stern (`*`) steht für eine beliebige Anzahl (auch Null) des vorhergehenden Zeichens. Durch Kombination mit den Klammern läßt sich die Suche weiter präzisieren. Um zum Beispiel nach allen Zeichenfolgen zu suchen, die mit einem beliebigen Buchstaben beginnen und mit „ist" enden, und dabei auch die Zeichenfolge „ist" selbst einzuschließen, geben Sie folgendes ein:

```
/[a-z]*ist
```

Zeichenfolgen ersetzen

Wenn Sie eine Zeichenfolge im gesamten Text durch eine andere ersetzen wollen, geben Sie den Suchbefehl in folgendem Format ein:

```
:g/suchtext/s//ersatztext/g
```

Die Eingabe wird mit „Return“ abgeschlossen. Alle im vorherigen Abschnitt beschriebenen Sonderzeichen können auch beim Ersetzen von Zeichenfolgen verwendet werden.

Um die Zeichenfolge „Computer“ durch „Workstation“ zu ersetzen, würden Sie also folgendes eingeben:

```
:g/Computer/s//Workstation/g
```

Wenn Sie nicht sicher sind, ob Sie eine Zeichenfolge überall ersetzen wollen, beenden Sie den Befehl mit `gc` statt mit `g`. In diesem Fall werden Sie bei jeder gefundenen Textstelle gefragt, ob die Zeichenfolge tatsächlich ersetzt werden soll. Sie können dann mit `y` (ja) oder `n` (nein) antworten.

```
:g/Computer/s//Workstation/gc
```

Hinweis – Wenn der Suchlauf mit Bestätigung durchgeführt wird, können Sie ihn mit „Ctrl-C“ abbrechen.

Zeilenwechsel

Mit dem Befehl `G` springen Sie zur letzten Zeile einer Datei. Um in eine beliebige andere Zeile zu wechseln, geben Sie vor dem Befehl die Nummer der gewünschten Zeile ein. Mit `1G` setzen Sie den Cursor also in die erste Zeile der Datei.

Wenn Sie die Datei `test` beispielsweise bis Zeile 51 bearbeitet und die Arbeitssitzung danach beendet haben, geben Sie nach dem nächsten Öffnen einfach `51G` ein und können dann an der Stelle weiterarbeiten, an der Sie zuvor aufgehört hatten.

Dateien verknüpfen

Mit folgendem Befehl können Sie den Inhalt einer Datei an einer beliebigen Stelle in die aktuell bearbeitete Datei einsetzen:

```
:zeile# r dateiname
```

Wenn Sie keine Zeilennummer angeben, wird die Datei an der aktuellen Cursorposition eingefügt.

Um die Datei `orwell` nach Zeile 84 der Datei `test` einzusetzen, würden Sie also folgendes eingeben:

```
:84 r orwell
```

Sie könnten statt dessen auch den Cursor in Zeile 84 setzen und folgenden Befehl eingeben:

```
:r orwell
```

Mehrere Dateien bearbeiten

Sie können mit `vi` auch mehrere Dateien in einer Editorsitzung bearbeiten. Um beispielsweise zwischen der Datei `test` und einer Datei namens `orwell`: zu wechseln, gehen Sie folgendermaßen vor:

1. **Geben Sie `:w` ein, und drücken Sie „Return“, um die aktuellen Änderungen an der Datei `test` zu speichern.**
2. **Geben Sie `:n orwell` ein, und drücken Sie „Return“, um die Datei `orwell` zu laden.**
3. **Bearbeiten Sie die Datei `orwell`, und speichern Sie die Änderungen.**
4. **Danach haben Sie folgende Möglichkeiten:**
   - `vi` beenden durch Eingabe von `:q` und „Return“.
   - Rückkehr zu `test` durch Eingabe von `:n #` und „Return“.
   - Wechseln zwischen den Dateien mit `:n #`.

Dateien nacheinander bearbeiten

Wenn Sie mehrere Dateien nacheinander bearbeiten wollen, geben Sie beim Aufruf von `vi` die Namen der Dateien an:

```
$ vi test orwell
```

Die Dateien werden in der Reihenfolge geladen, in der sie in der Befehlszeile aufgeführt sind. In unserem Beispiel erscheint also zuerst `test`. Wenn Sie mit der Bearbeitung dieser Datei fertig sind, geben Sie `:n` ein, um zur Datei `orwell` zu wechseln. Wenn Sie zur nächsten Datei übergehen wollen, ohne die Änderungen in der aktuellen Datei zu speichern, geben Sie `:n!` ein.

Haben die zu bearbeitenden Dateien ähnliche Namen (z. B. `test1`, `test2`, `test3`), können Sie bei der Eingabe der Dateinamen Platzhalter verwenden:

```
$ vi test*
```

In diesem Fall werden die Dateien in alphabetischer Reihenfolge geladen.

Zeilen zwischen Dateien kopieren

Um Zeilen aus einer Datei in eine andere Datei zu kopieren, gehen Sie wie folgt vor:

1. **Bearbeiten Sie die erste Datei.**
2. **Kopieren Sie die gewünschten Zeilen in einen benannten Puffer. Verwenden Sie hierzu den Befehl `yank`. Um z. B. 10 Zeilen in Puffer a zu speichern, geben sie `a10y` ein.**
3. **Wechseln Sie zur nächsten Datei (hier ist es die Datei `orwell`), ohne `vi` zu verlassen:**

```
:n orwell
```

4. **Setzen Sie den Inhalt des benannten Puffers in die Datei ein. Verwenden Sie hierzu den Befehl `put`. Um beispielsweise den Inhalt von Puffer *a* unterhalb der aktuellen Cursorposition einzusetzen, geben Sie `ap` ein.**

Da der Inhalt benannter Puffer verlorengeht, wenn Sie `vi` beenden, müssen Sie Prozeduren wie die zuvor beschriebene immer während der aktuellen Arbeitssitzung abschließen.

`vi`-*Parameter setzen*

`vi` unterstützt eine Reihe von Optionen, mit denen Sie das Verhalten des Editors beeinflussen können. Um diese Optionen während einer Arbeitssitzung anzuzeigen, geben Sie folgenden Befehl ein:

```
:set all
```

Schließen Sie die Eingabe mit „Return“ ab.

Neustart nach einem Systemabsturz

Wenn Ihr System abstürzt, besteht die Gefahr, daß der Inhalt des Dateipuffers verlorengeht. Der Pufferinhalt läßt sich jedoch durch einen Neustart von `vi` mit der Option `-r` häufig ganz oder teilweise wiederherstellen:

```
vi -r dateiname
```

dateiname ist der Name der Datei, die zum Zeitpunkt des Systemabsturzes bearbeitet wurde. In der Regel werden Sie nach dem Wiederhochfahren des Systems per elektronische Post informiert, daß eine Recover-Datei vorhanden ist.

 6

Befehlsübersicht

In der folgenden Tabelle sind die wichtigsten `vi`-Befehle zusammengefaßt.

Tabelle 6-1 `vi`-Befehle

Befehl	Beschreibung
`vi` *starten*	
`vi` *dateiname*	Öffnet eine Datei oder erstellt eine neue Datei
`vi`	Öffnet eine neue Datei ohne Namen
`vi -r` *dateiname*	Stellt eine Datei nach einem Systemabsturz wieder her
`view` *dateiname*	Öffnet eine Nur-Lese-Datei
Cursorbefehle	
h	Ein Zeichen nach links
j	Eine Zeile nach unten
k	Eine Zeile nach oben
l	Ein Zeichen nach rechts
w	Zum nächsten Wort oder Satzzeichen
W	Zum nächsten Wort
b	Zum vorhergehenden Wort oder Satzzeichen
B	Zum vorhergehenden Wort
e	An das Ende des aktuellen Worts
Return	Eine Zeile nach unten
Rücktaste	Ein Zeichen nach links
Leertaste	Ein Zeichen nach rechts
H	Zur ersten Bildschirmzeile
M	Zur Mitte des Bildschirms
L	Zur letzten Bildschirmzeile
„Ctrl-F“	Zur nächsten Bildschirmseite
„Ctrl-D“	Eine halbe Bildschirmseite nach unten
„Ctrl-B“	Zur vorhergehenden Bildschirmseite
„Ctrl-U“	Eine halbe Bildschirmseite nach oben

Tabelle 6-1 `vi`-Befehle (Fortsetzung)

Befehl	Beschreibung
Texteingabe	
a	Zeichen rechts vom Cursor einfügen
A	Zeichen am Zeilenende einfügen
i	Zeichen links vom Cursor einfügen
I	Zeichen am Zeilenanfang einfügen
o	Zeile unter dem Cursor einfügen
O	Zeile über dem Cursor einfügen
Textänderung	
cw	Wort rechts vom Cursor ändern
c	Zeile ändern
C	Zeile ab Cursorposition ändern
s	Zeile bis Cursorposition ändern
r	Aktuelles Zeichen ersetzen
`r` Return	Zeile teilen
J	Aktuelle und darunterstehende Zeile zusammenführen
xp	Aktuelles und rechts davon stehendes Zeichen vertauschen
~	Groß-/Kleinschreibung umkehren
u	Letzten Befehl rückgängig machen
U	Alle Änderungen an der aktuellen Zeile aufheben
:u	Letzten `ex`-Befehl rückgängig machen
Löschen	
x	Aktuelles Zeichen löschen
X	Zeichen links vom Cursor löschen
dw	Wort rechts vom Cursor ganz oder teilweise löschen
dd	Aktuelle Zeile löschen
D	Zeile ab Cursorposition löschen
dG	Dateiinhalt ab Cursorposition löschen

Tabelle 6-1 `vi`-Befehle (Fortsetzung)

Befehl	**Beschreibung**
d1G	Dateiinhalt bis Cursorposition löschen
:5,10 d	Zeilen 5-10 löschen
Kopieren und Bewegen	
yy	Zeile kopieren
Y	Zeile kopieren
p	Pufferinhalt unter der aktuellen Zeile einsetzen
P	Pufferinhalt über der aktuellen Zeile einsetzen
:1,2 co 3	Zeilen 1-2 kopieren und nach Zeile 3 einsetzen
:4,5 m 6	Zeilen 4-5 ausschneiden und nach Zeile 6 einsetzen
Zeilennummern	
:set nu	Zeilennummern anzeigen
:set nonu	Zeilennummern ausblenden
	Groß-/Kleinschreibung
:set ic	Groß-/Kleinschreibung bei Suchläufen ignorieren
:set noic	Groß-/Kleinschreibung bei Suchläufen beachten
Zeilenwechsel	
G	Zur letzten Zeile springen
1G	Zur ersten Zeile springen
21G	Zu Zeile 21 springen
Suchen und Ersetzen	
/*zeichenfolge*	Vorwärts nach *zeichenfolge* suchen
?*zeichenfolge*	Rückwärts nach *zeichenfolge* suchen
n	Nächste *zeichenfolge* in Suchrichtung finden
N	Vorhergehende *zeichenfolge* in Suchrichtung finden
:g/*suche*/s//*ersetze*/g	*suche* im gesamten Text durch *ersetze* ersetzen
Bildschirm	
„Ctrl-L“	Bildschirminhalt neu zeichnen

Tabelle 6-1 `vi`-Befehle (Fortsetzung)

Befehl	**Beschreibung**
	Dateien verknüpfen
`:r` *dateiname*	*dateiname* an Cursorposition einsetzen
`:34 r` *dateiname*	*dateiname* nach Zeile 34 einsetzen
Speichern und Beenden	
:w	Änderungen speichern
:w *dateiname*	Änderungen in *dateiname* speichern
:wq	Änderungen speichern und Arbeitssitzung beenden
ZZ	Änderungen speichern und Arbeitssitzung beenden
:q!	Arbeitssitzung ohne Speichern beenden



Elektronische Post



Zu SunOS gehört das Programm `mailx`, mit dem Sie Nachrichten über elektronische Post (*Email*) versenden und empfangen können. `mailx` bietet Funktionen zum Erstellen, Anzeigen, Speichern und Löschen von Nachrichten und kann, da es sich nicht um ein fenstergesteuertes Programm handelt, auf jedem Terminal ablaufen. Außer den Funktionen von `mailx` wird in diesem Kapitel auch beschrieben, wie Sie eigene Alias-Adressen definieren.

Hinweis – Wenn Sie sich in der OpenWindows-Umgebung befinden und das Symbol der Anwendung „Post" im Arbeitsbereich angezeigt wird, sollten Sie die Anwendung beenden, bevor Sie die Beispiele in diesem Kapitel durcharbeiten. Sie können Nachrichten zwar aus einer Kommando- oder UNIX-Shell versenden, ohne daß es zu Konflikten zwischen `mailx` und „Post" kommt, wenn Sie jedoch Nachrichten lesen, speichern oder löschen, beeinflußt das den „Briefkasten" der Anwendung „Post", und das System gibt unter Umständen Fehlermeldungen und Warnhinweise aus.

 7

Grundlagen

In diesem Abschnitt lernen Sie die wichtigsten Funktionen von `mailx` in einer Beispielsitzung kennen. Wenn Sie mehr über `mailx` wissen möchten, sollten Sie auch die darauffolgenden Abschnitte lesen, in denen weitere Befehle und Funktionen vorgestellt werden.

Um eine Nachricht an einen Benutzer schicken zu können, benötigen Sie dessen Login-Namen und den Hostnamen seines Systems. Arbeitet der Empfänger am gleichen System wie Sie, reicht der Login-Name aus. Jeder Benutzer verfügt über einen eigenen *Briefkasten*, der sich normalerweise im Verzeichnis `/var/mail/`*benutzername* befindet. (*benutzername* steht hier für den Login-Namen.)

`mailx` meldet, wenn eine Nachricht eintrifft, und legt die Nachricht im Briefkasten ab. Gelesene Nachrichten werden automatisch in einer Datei namens `mbox` gespeichert, die sich ebenfalls in Ihrem Home-Verzeichnis befindet.

`mailx` *starten*

Um `mailx` zu starten, geben Sie folgenden Befehl in die Befehlszeile ein und drücken „Return“:

```
$ mailx
```

Liegen keine neuen Nachrichten vor, gibt das System folgende Meldung aus:

```
Keine Post für benutzername
$
```

benutzername steht hier für Ihren Login-Namen.

Beispielnachricht senden

Damit Sie sehen, wie `mailx` funktioniert, sollten Sie als erstes eine Nachricht an sich selbst schicken. Dazu geben Sie den Befehl `mailx` noch einmal ein, diesmal aber zusammen mit Ihrer Postadresse (Login-Name und Hostname). In den folgenden Beispielen werden der Login-Name `calvin` und der Host-

name hobbes verwendet. Bei der Eingabe der Postadresse verbinden Sie diese beiden Elemente mit dem Zeichen @ (hier also zu calvin@hobbes). In einem lokalen Netzwerk genügt unter Umständen auch der Login-Name. Wenn Sie nicht wissen, was Sie auf Ihrem System eingeben müssen, fragen Sie Ihren Systemverwalter.

```
$ mailx calvin@hobbes
```

Das Programm zeigt daraufhin die Eingabeaufforderung Subject: an:

```
$ mailx calvin@hobbes
Subject:
```

Wenn Sie wollen, können Sie hier einen kurzen Kommentar zum Anlaß der Nachricht eingeben. (Die Eingabe muß mit „Return“ abgeschlossen werden.) Danach geben Sie den Nachrichtentext ein. Drücken Sie nach jeder Zeile „Return“. Tippfehler können nur in der aktuellen Zeile und nur mit der Rückt-aste gelöscht werden.

Um eine Leerzeile einzufügen, drücken Sie zweimal „Return“.

```
$ mailx calvin@hobbes
Subject: Interstellare Ausflüge

Lieber Calvin,

Raumfahrer Spiff begibt sich heute wieder auf eine
Expedition zum Planeten Zorg. Kommst du mit?

Bis bald,

Calvin
```

Um die Nachricht abzuschicken, drücken Sie nach Eingabe der letzten Zeile „Return“ und danach die Tastenkombination „Ctrl-D“. Die Nachricht wird abgeschickt, und auf dem Bildschirm erscheint wieder die Shell-Eingabe-aufforderung.

Beispielnachricht lesen

Um zu prüfen, ob neue Nachrichten eingetroffen sind, geben Sie den Befehl `mailx` noch einmal ein. Daraufhin werden etwa folgende Meldungen angezeigt:

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Betätigen Sie ?
für Hilfe.
"/var/mail/calvin": 2 Nachrichten 1 neu
 U  2 hal@ungewiss   Fri Feb 14 12:01   14/318 Finanzlage
>N  1 calvin@hobbes  Mon Feb 17 08:12   21/453 Interstellare
&
```

Die erste Zeile enthält Versionshinweise zu `mail`. In der zweiten Zeile ist Ihr Briefkasten angegeben (normalerweise `/var/mail/`*benutzername*), in dem die eingehende Post abgelegt wird. In den darauffolgenden Zeilen erscheinen die Kopfzeilen der im Briefkasten enthaltenen Nachrichten. Ein „N" am Anfang der Zeile gibt an, daß es sich um eine neue Nachricht handelt. Steht am Anfang der Zeile ein „U", heißt das, daß die Nachricht zwar schon während der letzten Arbeitssitzung eingetroffen ist, aber noch nicht gelesen wurde. (Auf diese Informationen wird im Abschnitt „Nachrichten lesen" noch ausführlich eingegangen.)

Jeder Nachricht wird beim Empfang eine Nummer zugewiesen. Calvins Brief an sich selbst trägt beispielsweise die Nummer `1`.

Um eine Nachricht zu lesen, geben Sie nach der `mailx`-Eingabeaufforderung die Nachrichtennummer ein. Zum Beispiel:

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Betätigen Sie ?
für Hilfe.
“/var/mail/calvin”: 1 Nachrichten 1 neu
>N  1 calvin@hobbes   Fri Jul 14 12:01 21/453 Interstellare
& 1

To: calvin@hobbes
From: calvin@hobbes
Subject: Interstellare Ausflüge

Lieber Calvin,

Raumfahrer Spiff begibt sich heute wieder auf eine
Expedition zum Planeten Zorg. Kommst du mit?

Bis bald,
Calvin

&
```

`mailx` *beenden*

Sie können `mailx` mit zwei Befehlen verlassen: `q` oder `x`.

Wenn Sie `q` eingeben und „Return“ drücken

```
& q
```

erscheint die folgende oder eine ähnliche Meldung:

`1 Nachricht gesichert in` *home_verzeichnis*`/mbox`.

home_verzeichnis steht für den Pfadnamen zu Ihrem Home-Verzeichnis.

Wenn Sie `mailx` mit dem Befehl `q` beenden, werden alle gelesenen Briefe aus dem Briefkasten in die Datei `mbox` in Ihrem Home-Verzeichnis übertragen. `mailx` speichert darüber hinaus auch alle Änderungen, die Sie während der aktuellen Arbeitssitzung vorgenommen haben.

Wenn Sie `x` eingeben und „Return“ drücken

```
& x
```

werden Änderungen *nicht* gespeichert und die gelesenen Nachrichten verbleiben im Briefkasten.

Nachrichten lesen

Sind seit der letzten Arbeitssitzung neue Nachrichten eingetroffen, zeigt `mailx` nach der Anmeldung eine der folgenden Meldungen an:

`Es ist Post für Sie da`

oder

`Es ist neue Post da`

Sind keine neuen Nachrichten vorhanden, erscheint folgende Meldung:

`Keine Post für` *benutzername*

Um den Inhalt des Briefkastens anzuzeigen, geben Sie `mailx` ein und drücken „Return“:

```
$ mailx
Mail version 4.0 Thu Jan 16 12:59:09 PST 1992  Betätigen Sie ?
für Hilfe.
“/var/mail/calvin”: 4 Nachrichten 1 neu 2 ungelesen
    1 calvin@hobbes Fri Feb 14 12:01 21/453 Interstellare
 U  2 jon@garfield  Fri Feb 14 18:31 19/353 zwei lasagne
 U  3 laurel@hardy  Sat Feb 15 10:22 24/557 meeting um 2
>N  4 donald@pluto  Sun Feb 16 23:59 14/280 Knochenbestell

&
```

In der ersten Zeile werden Informationen zu `mailx` selbst (Versionsnummer und Datum) und ein Hinweis zum Abrufen von Online-Hilfe (`Betätigen Sie ? für Hilfe`) angezeigt.

In der nächsten Zeile zeigt `mailx` den Pfadnamen des Briefkastens, die Anzahl eingetroffener Nachrichten und den Status dieser Nachrichten an.

Danach folgt eine numerierte Liste mit den aktuell im Briefkasten enthaltenen Nachrichten. Zu jeder Nachricht sind folgende Informationen angegeben:

- *Status*: Gibt an, ob die Nachricht neu ist (N), noch nicht gelesen wurde (U) oder bereits gelesen ist (kein Symbol). Die aktuelle Nachricht ist mit dem Zeichen > gekennzeichnet. Gelöschte Nachrichten sind mit einem Stern (*) markiert.
- *Nummer*: Nachrichten werden in der Reihenfolge des Eingangs numeriert.
- *Absender*: Name und (meist) Systemname des Benutzers, von dem die Nachricht stammt.
- *Zeit*: Datum und Zeitpunkt, zu dem die Nachricht abgeschickt wurde.
- *Größe*: Anzahl der Zeilen und Zeichen in der Nachricht.
- *Thema*: Betreff-Zeile (Subject) der Nachricht.

Wenn die Liste nicht vollständig auf den Bildschirm paßt, können Sie mit folgenden Befehlen vor- und zurückblättern:

- `z` – Zeigt die nächste Bildschirmseite an.
- `h` – Zeigt die vorhergehende Bildschirmseite an.
- `h` – Zeigt die Kopfzeilenliste wieder an.

Um die aktuelle Nachricht (>) zu lesen, drücken Sie „Return“. Wenn Sie danach ein zweites Mal „Return“ drücken, wird die nächste Nachricht angezeigt usw. Um eine beliebige Nachricht anzuzeigen, geben Sie ihre Nummer ein und drücken dann „Return“.

Nachrichten löschen und wiederherstellen

Normalerweise werden gelesene Nachrichten in der Datei `mbox` gespeichert, wenn Sie `mailx` verlassen. Wird eine Nachricht nicht mehr benötigt, können Sie sie aber auch aus dem Briefkasten löschen.

Um die zuletzt gelesene Nachricht zu löschen, geben Sie einfach `d` ein. Soll eine bestimmte Datei gelesen werden, verwenden Sie folgendes Eingabeformat:

`d` *nummer*

Mit folgendem Befehl löschen Sie die zweite Nachricht aus dem Briefkasten:

```
& d 2
```

Sie können auch mehrere Nachrichten gleichzeitig löschen. Um die Nachrichten 1 *und* 3 zu löschen, geben Sie folgendes ein:

```
& d 1 3
```

Mit folgendem Befehl löschen Sie die Nachrichten 1 *bis* 3:

```
& d 1-3
```

Wenn Sie eine Nachricht wiederherstellen wollen, müssen Sie das *vor* Beendigung des Programms tun. Dazu verwenden Sie folgenden Befehl:

`u` *nummer*

Die Eingabe wird mit „Return" abgeschlossen. Mit dem nächsten Befehl stellen Sie die zweite Nachricht wieder her:

```
& u 2
```

Wenn Sie `u` ohne eine Nummer eingeben, wird der letzte Löschbefehl rückgängig gemacht. Wenn Sie zuletzt `d 2-5` eingegeben haben, stellen Sie mit `u` also die Nachrichten `2`, `3`, `4` und `5` wieder her.

Sobald Sie `mailx` mit dem Befehl `q` beenden, sind alle gelöschten Dateien endgültig verloren. Beenden Sie das Programm dagegen mit dem Befehl `x`, bleibt der Inhalt des Briefkastens unverändert und Sie können gelöschte Nachrichten auch in der nächsten Arbeitssitzung wiederherstellen.

Nachrichten drucken

Um eine Nachricht zu drucken, leiten Sie sie an einen Druckbefehl weiter:

|*nummer* `lp`

(Das Zeichen | wird auch als *Pipe* bezeichnet.) Um eine Kopie der Nachricht 2 zu drucken, geben Sie folgenden Befehl ein und drücken „Return“:

```
& |2 lp
```

Wenn Sie keine Nachrichtennummer angeben, sendet `mailx` die aktuelle Nachricht an den Drucker. Weitere Hinweise zur Verwendung des Pipe-Symbols finden Sie im Abschnitt „Befehlsausgabe umleiten“ in Kapitel 2, „SunOS-Befehle“.

Nachrichten senden

Um eine Nachricht mit `mailx` abschicken zu können, benötigen Sie den Login-Namen und den Systemnamen des Empfängers. (Befindet sich der Empfänger auf dem gleichen System, genügt der Login-Name.) Diese Informationen können Sie mit den Befehlen `who`, `finger` oder `rusers` abrufen.

Mit `who` zeigen Sie alle Benutzer an, die bei Ihrem Datei-Server angemeldet sind. Die Ausgabeliste enthält Benutzernamen, Terminaltyp und Datum und Zeitpunkt der Anmeldung:

```
$ who
    calvin      tty15       Feb 20 10:22
    donald      tty04       Feb 20 10:37
    jon         tty07       Feb 20 11:49
    tom         tty06       Feb 20 12:02
```

Der Befehl `finger` gibt eine etwas erweiterte Liste aus, deren genauer Inhalt durch den Systemverwalter festgelegt wird. Die folgende Abbildung zeigt ein Beispiel:

```
$ finger
    Login     Name           TTY      Idle     When
    calvin    Calvin Crump   tty15     43      Thu 10:22
    donald    Donald Doodle  tty04             Thu 10:37
    jon       Jon James      tty07     12      Thu 11:49
    tom       Tom Tool       tty06     22      Thu 12:02
```

Mit dem Befehl `rusers` stellen Sie fest, welche Benutzer auf welchen Systemen des lokalen Netzwerks angemeldet sind. Eine genauere Beschreibung finden Sie in Kapitel 9, „Arbeiten im Netzwerk“.

Wenn Sie die benötigten Informationen gefunden haben, können Sie die Nachricht abschicken:

1. Geben Sie `mailx` und die Adresse des Empfängers ein:

```
$ mailx benutzer@system
```

benutzer steht für den Login-Namen des Empfängers und *system* für den Hostnamen seines Systems.

- Wenn Sie `mailx` schon gestartet haben, genügt es, wenn Sie den Befehl `m` und die Empfängeradresse eingeben:

```
& m benutzer@system
```

- Wenn Sie eine Nachricht an mehrere Empfänger schicken wollen, trennen Sie die einzelnen Adressen durch Kommas oder Leerzeichen. Zum Beispiel:

```
$ mailx donald@pluto jon@garfield tom@jerry
```

oder

```
$ mailx donald@pluto,jon@garfield,tom@jerry
```

2. **Drücken Sie „Return“. Das Programm fordert Sie zur Eingabe einer Betreff-Zeile auf. Geben Sie einen kurzen Kommentar ein, und drücken Sie noch einmal „Return“.**

3. **Geben Sie den Text der Nachricht ein, und beenden Sie jede Zeile mit „Return“.**
   Wenn Sie keine Zeilenschaltungen eingeben, wird der Text zwar auf dem Bildschirm umbrochen, intern aber wie eine einzige Zeile behandelt.

Hinweis – Jede Textzeile kann maximal 256 Zeichen lang sein. Wenn Sie diese Grenze überschreiten, wird der Bildschirm „eingefroren“, und Sie müssen die Eingabe mit „Ctrl-C“ abbrechen.

---

4. **Drücken Sie nach Eingabe der letzten Zeile „Return“ und dann „Ctrl-D“, um die Nachricht abzuschicken.**

Nicht zustellbare Nachrichten

Wenn Sie eine falsche Empfängeradresse angeben, reagiert das System mit folgender Meldung und stellt die Nachricht zurück in den Briefkasten:

benutzer@system`...User unknown`

Wenn Sie `mailx` das nächste Mal starten, erscheint folgende Meldung:

```
N 1 Mailer-Daemon Fri Jan 3 11:13 8/49 Returned mail: User unknown
```

Nicht zugestellte Nachrichten werden außerdem in eine Datei namens `dead.letter` in Ihrem Home-Verzeichnis kopiert.

Eingabe abbrechen

Um die Eingabe einer Nachricht abzubrechen, drücken Sie zweimal „Ctrl-D“.

Kopien und Blindkopien

Bevor Sie eine Nachricht abschicken, können Sie festlegen, daß außer dem Empfänger noch andere Benutzer eine Kopie oder Blindkopie der Nachricht erhalten sollen. (In einer Blindkopie sind die Adressen der weiteren Empfänger nicht sichtbar.)

Nachrichtenkopien können auch an die eigene Adresse gesendet werden.

Es gibt drei Methoden, Kopien einer Nachricht zu versenden.

- Wenn Nachrichten standardmäßig immer mit einer oder mehreren Kopien verschickt werden sollen, öffnen Sie die Datei `.mailrc` in Ihrem Home-Verzeichnis mit einem Texteditor und nehmen folgende Zeile in die Datei auf:

```
set askcc
```

  Danach fordert `mailx` Sie bei Eingabe einer Nachricht automatisch dazu auf, die Adressen eines oder mehrerer Kopienempfänger einzugeben. Die Eingabeaufforderung `Cc:` erscheint nach der Aufforderung `Subject:`.

- Wenn Sie Kopien nur gelegentlich verschicken, können Sie die Adressen der Empfänger direkt nach der Eingabe des Nachrichtentexts angeben. Dazu geben Sie in die letzte Zeile einen Befehl in folgendem Format ein:

  `~c` *adresse(n)*

  Danach drücken Sie nicht noch einmal „Return“, sondern gleich „Ctrl-D“. Wenn Sie mehrere Kopien versenden wollen, trennen Sie die Adressen durch Leerzeichen. Zum Beispiel:

```
~c donald@pluto daniel@saturn doris@venus
```

- Mit dem Befehl `~h`, der wie alle Tilde-Befehle in eine separate Zeile eingegeben werden muß, zeigen Sie nacheinander die Eingabeaufforderungen `To:`, `Subject:`, `Cc:` und `Bcc:` (für Blindkopie) an. Enthält die Zeile bereits einen Eintrag, können Sie ihn überschreiben.

Hinweis – Eine Übersicht aller Tilde-Befehle finden Sie im Abschnitt „Tilde-Befehle“ am Ende dieses Kapitels.

Nachrichten oder Dateien übernehmen

Bei der Eingabe einer Nachricht können Sie eine andere Nachricht oder eine Textdatei in den Nachrichtentext kopieren.

Nachrichten einsetzen

Um eine vorhandene Nachricht in eine Nachricht zu kopieren, geben Sie folgenden Befehl ein:

`~m` *nummer*

nummer steht für die Nummer der Nachricht, die eingesetzt werden soll. Um beispielsweise eine Nachricht mit einer Kopie der Nachricht 3 aus der Briefkastenliste an einen anderen Benutzer zu schicken, gehen Sie so vor:

1. **Geben sie** `~m 3` **in eine neue Zeile ein, und drücken Sie „Return“.**
2. `mailx` **zeigt die Meldung** `Interpoliere: 3 (Fortfahren)` **an.**
3. **Der Text von Nachricht 3 wird nicht angezeigt, erscheint aber später auf dem Bildschirm des Empfängers. Sie können die Nachricht nach** `(Fortfahren)` **fortsetzen oder sie in der vorliegenden Form abschicken.**
4. **Mit dem Befehl** `~p` **zeigen Sie den Text der eingesetzten Nachricht an.**

Dateien einsetzen

Mit folgendem Befehl kopieren Sie eine Textdatei in Ihre Nachricht:

`~r` *dateiname*

Um die Datei `entwurf` einzusetzen, geben Sie also folgendes ein:

```
~r outline
```

Nachrichten beantworten

Um eine Nachricht zu beantworten, geben Sie folgenden Befehl in die `mailx`-Befehlszeile ein:

`r` *nummer*

Wenn Sie die Nummer weglassen, wird die aktuelle Nachricht beantwortet. Um Nachricht 2 zu beantworten, geben Sie also folgendes ein:

```
& r 2
```

`mailx` setzt daraufhin die Absenderadresse in das Adreßfeld und kopiert den Inhalt der Betreff-Zeile in eine neue Zeile mit dem Titel `Re: Subject:`. Danach geben Sie den Text der Nachricht ein und schicken die Nachricht ab.

Mit `R` senden Sie die Antwort nicht nur an den Absender, sondern auch an alle weiteren Empfänger der ursprünglichen Nachricht. Um eine „Nachrichten-inflation“ zu vermeiden, sollten Sie diesen Befehl jedoch sparsam verwenden.

Hinweis – Wenn Sie eine Kopie der ursprünglichen Nachricht in die Antwort aufnehmen wollen, geben Sie den Befehl `~m` ohne Nachrichtennummer ein.

Nachrichten speichern

Wenn Sie eine Nachricht voraussichtlich noch benötigen, können Sie sie in einer Datei speichern und bei Bedarf aus der Datei lesen. Dabei gibt es zwei Möglichkeiten: Sie können die Nachricht an das Ende einer Textdatei stellen oder sie in einer speziellen Datei ablegen, die als *Ordner* bezeichnet wird.

`mailx` unterscheidet zwischen dem *Speichern* und dem *Kopieren* einer Nachricht. Beim Speichern wird die Nachricht aus dem Briefkasten in eine Datei oder einen Ordner übertragen. Beim Kopieren verbleibt eine Kopie der Nachricht im Briefkasten.

Nachrichten in Dateien speichern

Mit folgendem Befehl speichern Sie eine Nachricht in einer Textdatei:

`s` *nummer dateiname*

nummer steht für die Nachrichtennummer und *dateiname* für die Datei, in der die Nachricht gespeichert werden soll. Zum Beispiel:

```
& s 3 ~/memos/finanzen
```

Damit speichern Sie Nachricht 3 in einer Datei namens `finanzen` in Ihrem Home-Verzeichnis. (Die Tilde kann nur in der C-Shell verwendet werden.) Ist die Datei bereits vorhanden, wird die Nachricht am Dateiende eingefügt. Existiert sie noch nicht, wird sie von `mailx` angelegt.

Sie können auch mehrere Nachrichten in einer Datei speichern. Zum Beispiel:

```
& s 3 5-8 ~/memos/finanzen
```

Mit diesem Befehl speichern Sie die Nachrichten 3, 5, 6, 7 und 8 in der Datei `~/memos/finanzen`.

Beim Speichern wird die Nachricht aus dem Briefkasten entfernt. Gespeicherte Nachrichten sind in der Kopfzeilenliste mit einem Stern (*) gekennzeichnet.

Soll eine Kopie der Nachricht im Briefkasten verbleiben, kopieren Sie die Nachricht mit dem Befehl `c` in die Datei:

```
& c 3 ~/memos/finanzen
```

Nachrichten in Ordnern speichern

Wenn Sie Nachrichten in Mail-Ordnern speichern, brauchen Sie beim Speichern oder Kopieren keine vollständigen Pfadnamen anzugeben. Ordner sind spezielle Dateien, die in einem eigenen Verzeichnis abgelegt werden.

Durch Anlegen entsprechender Ordner können Sie Ihre Nachrichten auch nach Thema oder Absender geordnet speichern.

Ordnerverzeichnis vorbereiten

Um Nachrichten in Ordnern speichern zu können, müssen Sie zunächst ein Ordnerverzeichnis anlegen und den Pfad zu diesem Verzeichnis setzen. Dazu gehen Sie folgendermaßen vor:

1. **Legen Sie das Verzeichnis mit dem Befehl** `mkdir` **an.**
   Zum Beispiel:

```
$ mkdir Nachrichten
```

2. **Öffnen Sie die `mailx`-Konfigurationsdatei `.mailrc` in Ihrem Home-Verzeichnis, und setzen Sie die Variable `set folder` auf den vollständigen Pfadnamen des neuen Verzeichnisses.**
   Zum Beispiel:

```
set folder=/home/berlin/nina/Nachrichten
```

   In der C-Shell können Sie den Pfadnamen des Home-Verzeichnisses auch durch eine Tilde ersetzen:

```
set folder=~/Nachrichten
```

Die Änderungen an der Datei `.mailrc` werden beim nächsten Start von `mailx` wirksam.

Ordnernamen

Um Nachrichten in Ordnern zu speichern oder in Ordner zu kopieren, verwenden Sie die gleichen Befehle wie für Dateien, stellen vor den Ordnernamen aber ein Pluszeichen (+) statt eines Pfadnamens. Das Pluszeichen gibt an, daß es sich bei dem nachfolgenden Namen um eine Ordnerdatei handelt, die im Ordnerverzeichnis gespeichert ist.

Um Nachricht 3 in einem Ordner namens `projekte` zu speichern, geben Sie also folgendes ein:

```
& s 3 +projekte
```

Damit weisen Sie das Programm an, Nachricht 3 in `~/Nachrichten/projekte` zu speichern. Existiert der Ordner `projekte` noch nicht, wird er angelegt.

Um die Datei in den Ordner zu *kopieren*, geben Sie folgendes ein:

```
& c 3 +projekte
```

Nachrichten an Dateien oder Ordner senden

Um die Kopie einer Nachricht in einem Ordner abzulegen, geben Sie den Ordnernamen in das Feld `Cc:` oder `Bcc:` ein. Wenn Sie eine Kopie der Nachricht an eine Datei senden wollen, müssen Sie den vollständigen Pfad angeben.

Nachrichten aus Dateien oder Ordnern lesen

Mit folgendem Befehl zeigen Sie den Inhalt einer Datei an, in der Nachrichten gespeichert sind:

`mailx -f` *dateiname*

Um den Inhalt der Datei `~/memos/finanzen` anzuzeigen, würden Sie also folgendes eingeben:

```
$ mailx -f ~/memos/finanzen
```

Um den Inhalt eines Ordners zu lesen, geben Sie vor dem Dateinamen ein Pluszeichen statt des Pfads ein. Zum Beispiel:

```
$ mailx -f +projekte
```

Wenn Sie `mailx` in einer Datei oder einem Ordner starten, werden nur die Kopfzeilen der darin enthaltenen Nachrichten angezeigt. Um eine Nachricht zu lesen, geben Sie ihre Nummer ein und drücken „Return".

Wenn `mailx` bereits läuft, können Sie zwischen dem Briefkasten und den vorhandenen Ordnern hin- und herschalten. Mit folgendem Befehl zeigen Sie eine Liste der verfügbaren Ordner an:

```
& folder
```

Mit dem nächsten Befehl wechseln Sie aus dem Briefkasten in einen Ordner:

```
& folder +ordnername
```

Mit folgender Eingabe kehren Sie wieder zum Briefkasten zurück:

```
& %
```

Um in den vorhergehenden Ordner zu wechseln, geben Sie folgendes ein:

```
& #
```

`vi` *und* `mailx`

Sie können bei der Eingabe einer Nachricht auch den Texteditor `vi` aufrufen. In diesem Fall stehen Ihnen viele Korrektur- und Bearbeitungsfunktionen zur Verfügung, die `mailx` nicht bietet. Wenn Sie noch nicht mit `vi` gearbeitet haben, lesen Sie bitte Kapitel 6, „Textbearbeitung mit dem Editor vi“.

So erstellen Sie eine Nachricht mit `vi`:

1. **Geben Sie in der Shell oder in der `mailx`-Befehlszeile den Befehl `mailx` und die Adresse des Empfängers der Nachricht ein. Drücken Sie „Return“.**

2. **Geben Sie das Thema der Nachricht in die Zeile `Subject:` ein, und drücken Sie „Return“.**

3. **Geben Sie den Befehl `~v` am Anfang einer neuen Zeile ein. `vi` wird gestartet und öffnet eine leere Datei im Unterverzeichnis `/tmp` Ihres Home-Verzeichnisses.**

4. **Geben Sie den Text der Nachricht mit `vi` ein.**

5. **Beenden Sie `vi` mit dem Befehl `:wq` oder `ZZ`.**

`mailx` zeigt daraufhin im Nachrichtentext die Meldung `(Fortfahren):` an. Sie können die Eingabe entweder an dieser Stelle fortsetzen oder die Nachricht mit „Ctrl-D“ abschicken.

Alias-Adressen

Mit Hilfe einer *Alias-Adresse* oder *Adreßliste* können Sie Nachrichten an mehrere Benutzer senden, ohne die Postadresse jedes einzelnen Benutzers angeben zu müssen.

Nehmen wir an, Sie schicken in regelmäßigen Abständen Nachrichten an die Benutzer `hans@neptun`, `theo@mars` und `franz@jupiter`. Um nicht jedesmal drei Adressen eingeben zu müssen, erstellen Sie eine Alias-Adresse namens `amigos` und geben diese in der Adreßzeile an. Die Nachricht wird dann automatisch an alle drei Benutzer gesendet.

Alias-Adressen können in folgenden Dateien definiert werden:

- `.mailrc` (in Ihrem Home-Verzeichnis)
- `/etc/aliases`

Zwischen Alias-Adressen, die Sie in der Datei `.mailrc` definieren, und den Adressen in der Datei `/etc/aliases` bestehen Unterschiede, die in Tabelle 7-1 am Ende dieses Abschnitts zusammengefaßt sind.

Alias-Adressen in `.mailrc`

Die Alias-Adressen in `.mailrc` weisen folgende Besonderheiten auf:

- Sie sind *privat*, d. h. sie können nur von Ihnen verwendet werden. Wenn ein anderer Benutzer versucht, eine Nachricht an eine dieser Adressen zu senden, erhält er die Fehlermeldung `Unbekannter Benutzer`.
- Wenn Sie eine Nachricht verschicken, wird bei den Empfängern nicht die Alias-Adresse, sondern eine Liste aller Empfänger angezeigt. Für die Empfänger ist also nicht ersichtlich, daß Sie eine Alias-Adresse verwendet haben.

`.mailrc` befindet sich in Ihrem Home-Verzeichnis und enthält Konfigurationseinstellungen für `mailx` und die OpenWindows-Anwendung „Post“.

Um die Datei `.mailrc` zu bearbeiten, geben Sie folgendes ein:

```
$ vi ~/.mailrc
```

Hinweis – Sie können statt `vi` auch einen anderen Texteditor verwenden. Wenn Sie mit `vi` noch nicht vertraut sind, schlagen Sie bitte in Kapitel 6, „Textbearbeitung mit dem Editor vi“ nach.

Jede Alias-Adresse muß in eine separate Zeile eingegeben werden. Sie kann zwar länger als eine Zeile sein, darf aber keine Zeilenschaltungen enthalten. Jede Adresse besteht aus folgenden Elementen, die durch Leerzeichen getrennt werden:

- Dem Wort „alias“
- Dem Namen der Alias-Adresse (ein Wort)
- Den Adressen (Login- und Systemname) der Empfänger

Das nächste Beispiel zeigt zwei Alias-Adressen, eine mit drei und eine mit sechs Empfängern. Die zweite Adresse ist länger als eine Zeile, was aber problemlos ist, solange Sie bei der Eingabe keine Zeilenschaltungen verwenden.

```
alias amigos hans@neptun theo@mars franz@jupiter
alias kollegen jon@garfield tom@jerry donald@pluto
laurel@hardy nina@texas steve@london
```

Um eine Nachricht an eine Alias-Adresse zu senden, geben Sie in der Adreßzeile nur den Namen der Alias-Adresse an. Zum Beispiel:

```
$ mail amigos
Subject: Urlaub

Hat einer von euch Lust, einen kleinen Abstecher nach Rio
zu machen? Sagt mir Bescheid!
```

Auf den Systemen der Empfänger erscheint diese Nachricht folgendermaßen:

```
To: hans@neptun theo@mars franz@jupiter
Subject: Urlaub

Hat einer von euch Lust, einen kleinen Abstecher nach Rio
zu machen? Sagt mir Bescheid!
```

Alias-Adressen in `/etc/aliases`

Alias-Adressen in der Datei `/etc/aliases` haben folgende Eigenschaften:

- Sie sind *öffentlich*, d. h. sie können (im Format *alias_name@Ihr_systemname*) von jedem Benutzer des Netzwerks verwendet werden.
- Wenn Sie eine Nachricht an eine dieser Alias-Adressen schicken, wird der Alias auch bei den Empfängern angezeigt. Diese wissen also, daß Sie eine Alias-Adresse verwendet haben, aber nicht unbedingt, wer darin noch angegeben ist.

Alias-Adressen in `/etc/aliases` unterschieden sich auch im Format von den Adressen in `.mailrc`. Sie bestehen aus folgenden Elementen:

- Dem Namen des Alias, gefolgt von einem Doppelpunkt (`:`)
- Den Adressen (Login- und Systemname) der Empfänger, die durch Kommas getrennt werden. Dieser Adreßtyp kann auch länger als eine Zeile sein.

Um die Datei `/etc/aliases` bearbeiten zu können, müssen Sie sich unter dem Benutzernamen `root` anmelden. Wenn `root` paßwortgeschützt ist, müssen Sie das Paßwort kennen oder Ihren Systemverwalter bitten, die Änderungen für Sie vorzunehmen.

Um sich als `root` (Superuser) anzumelden, geben Sie folgendes ein:

```
$ su
Password:
#
```

Für den Superuser wird eine andere Eingabeaufforderung angezeigt.

Im folgenden Beispiel wird die Alias-Adresse `kollegen@texas` in die Datei `/etc/aliases` aufgenommen. (Die Abbildung zeigt, wie die Datei nach der Installation der Systemsoftware aussieht. Ihre Datei enthält unter Umständen andere Informationen.)

```
# vi /etc/aliases
##
#Aliases can have any mix of upper and lower case on the left-
#hand side,
#but the right-hand side should be proper case (usually lower)
#
#     >>>>>>>>>>The program "newaliases" will need to be run after
#      >> NOTE >>this file is updated for any changes to
#      >>>>>>>>>>show through to sendmail.
#
#@(#)aliases 1.10 89/01/20 SMI
##
# Following alias is required by the mail protocol, RFC 822
# Set it to the address of a HUMAN who deals with this system's
mail problems.
Postmaster: root

# Alias for mailer daemon; returned messages from our MAILER-
DAEMON
# should be routed to our local Postmaster.
MAILER-DAEMON: postmaster

# Aliases to handle mail to programs or files, eg news or vacation
# decode: "|/usr/bin/uudecode"
nobody: /dev/null

# Sample aliases:
# Alias for distribution list, members specified here:
#staff:wnj,mosher,sam,ecc,mckusick,sklower,olson,rwh@ernie

# Alias for distribution list, members specified elsewhere:
#keyboards: :include:/usr/jfarrell/keyboards.list

# Alias for a person, so they can receive mail by several names:
#epa:eric

#######################
# Local aliases below #
```

```
######################
kollegen@texas: jon@garfield tom@jerry donald@pluto
laurel@hardy nina@texas steve@london
:wq          (vi beenden und /etc/aliases speichern)
# exit       (Superuser-Modus beenden)
$
```

Sie können die Datei `/etc/aliases` auch mit einem anderen Texteditor als `vi` bearbeiten. Eine ausführliche Beschreibung der Funktionen und Befehle von `vi` finden Sie in Kapitel 6, „Textbearbeitung mit dem Editor vi“.

Zeilen, die mit dem Zeichen # beginnen, sind *Kommentare*. Das Kommentar-zeichen verhindert, daß Befehle ausgeführt werden, die nur zur Erläuterung dienen (wie hier die Beispiel-Adreßlisten).

Wenn Sie Alias-Adressen in die Datei aufnehmen, dürfen Sie die Zeilen also nicht mit einem solchen Zeichen beginnen lassen (außer, Sie wollen eine Adresse bewußt streichen).

Um eine Nachricht an eine der in `/etc/aliases` angegebenen Alias-Adressen zu senden, geben Sie in der Adreßzeile den Namen der Alias-Adresse und Ihren Systemnamen an. Zum Beispiel:

```
$ mail kollegen@texas
Subject: Meeting um 16:30

Die Finanzlage ist im vergangenen Monat noch prekärer geworden.
Wir müssen uns dringend nach neuen Einnahmequellen umsehen.
Bitte seid alle pünktlich um 16:30 zur Stelle.
```

Da die Adreßzeile hier nicht, wie bei den Alias-Adressen aus `.mailrc`, erweitert wird, sehen die Empfänger das gleiche wie der Absender:

```
To: kollegen@texas
Subject: Meeting 16:30

Die Finanzlage ist im vergangenen Monat noch prekärer geworden.
Wir müssen uns dringend nach neuen Einnahmequellen umsehen.
Bitte seid alle pünktlich um 16:30 zur Stelle.
```

Wenn Sie Nachrichten an eine Alias-Adresse dieses Typs schicken, müssen Sie den Namen des Systems angeben, auf dem die Adresse gespeichert ist. Wenn Sie auf dem System `riders` eine Alias-Adresse namens `easy` definieren, müssen Sie die Nachricht also an `easy@riders` schicken.

In Tabelle 7-1 sind die Unterschiede zwischen Alias-Adressen aus der Datei `.mailrc` und der Datei `/etc/aliases` zusammengefaßt.

Tabelle 7-1 Unterschiede zwischen Alias-Adressen in `.mailrc` und `/etc/aliases`

	.mailrc	**/etc/aliases**
Änderungen nur als `root` möglich?	nein	ja
Adreßzeilenformat	*alias*	*alias@systemname*
Empfängerliste für Empfänger sichtbar?	ja	nein
Empfängeradressen durch Kommas getrennt?	nein	ja
Alle Adressen in einer Zeile?	ja	nein
Verwendung durch andere Benutzer möglich?	nein	ja

Wenn Sie mehr über Alias-Adressen wissen möchten, geben Sie `man aliases` oder `man addresses` in die Befehlszeile ein.

Tilde-Befehle

Beim Erstellen einer Nachricht können Sie mit Hilfe von Tilde-Befehlen eine Reihe spezieller Programmfunktionen aufrufen. Diese Befehle bestehen immer aus einer Tilde (~) und einem einzelnen Zeichen. In der folgenden Tabelle sind die am häufigsten verwendeten Tilde-Befehle zusammengefaßt. Einige davon wurden in diesem Kapitel bereits behandelt.

Hinweis – Wenn Sie eine Tilde innerhalb des Nachrichtentexts verwenden wollen, müssen Sie zwei Tilde-Zeichen eingeben, von denen nur eine angezeigt wird.

Tabelle 7-2 Tilde-Befehle (`mailx`)

Befehl	Funktion
~!*befehl*	Führt einen Shell-Befehl aus.
~.	Simuliert die Eingabe von „Ctrl-D“ (Dateiende).
~?	Zeigt die verfügbaren Tilde-Befehle an.
`~b` *name*	Nimmt den angegebenen Benutzer in die Zeile `Bcc:` auf.
`~c` *name*	Nimmt den angegebenen Benutzer in die Zeile `Cc:` auf.
~d	Liest den Inhalt der Datei `dead.letter` in die aktuelle Nachricht.
`~f` *nummer*	Leitet die angegebene Nachricht weiter. Nur gültig beim Lesen einer Nachricht.
~h	Zeigt die Eingabeaufforderungen `Subject:`, `To:`, `Cc:` und `Bcc:` an.
`~m` *nummer*	Setzt den Text der angegebenen Nachricht in die aktuelle Nachricht ein. Nur gültig beim Lesen einer Nachricht.
~p	Gibt den eingegebenen Text als Meldung auf dem Bildschirm aus.
~q	Simuliert zweimalige Eingabe von „Ctrl-C“. Vorhandener Nachrichtentext wird in der Datei `dead.letter` gespeichert.
`~r` *dateiname*	Liest den Inhalt der angegebenen Datei in die aktuelle Nachricht.
`~s` *zeichenfolge*	Ändert die Betreff-Zeile in *zeichenfolge*.
`~t` *name*	Nimmt den angegebenen Namen in die Adreßzeile auf.
`~w` *dateiname*	Schreibt die aktuelle Nachricht ohne Kopfzeile in die angegebene Datei.
~x	Beendet `mailx`. Vorhandener Nachrichtentext wird nicht gespeichert.

Online-Hilfe zu `mailx`

Innerhalb einer `mailx` -Sitzung können Sie im Befehlsmodus durch Eingabe eines Fragezeichens eine Liste der in diesem Modus verfügbaren Befehle und im Eingabemodus mit `~?` eine Liste der verfügbaren Tilde-Befehle abrufen.

Wenn Sie mehr Informationen benötigen, geben Sie folgenden Shell-Befehl ein, um das Online-Dokument zu `mailx` anzuzeigen:

```
$ man mailx
```

Eine ausführliche Beschreibung von `mailx` finden Sie auch im *man Pages(1): User Commands* nach.

Drucken 8

Beim Drucken einer Datei greifen Sie auf den SunOS Druckdienst LP (*Line Printer*) zu. Da eine Beschreibung des gesamten Funktionsumfangs von LP den Rahmen dieses Handbuchs sprengen würden, werden hier nur die Funktionen behandelt, die für die Ausführung der wichtigsten Druckaufgaben benötigt werden:

- Druckanforderungen absetzen (eine Datei an einen Drucker senden)
- Druckerstatus ermitteln
- Druckanforderungen abbrechen

Eine ausführliche Beschreibung des Druckdienstes LP finden Sie im Handbuch *User Accounts, Printers, and Mail Administration*.

Druckanforderungen senden

Um eine Datei über die Befehlszeile zu drucken, senden Sie mit dem Befehl `lp` eine Druckanforderung an den Drucker. Daraufhin wird die Datei in die Warteschlange des Druckers gestellt, und auf dem Bildschirm erscheint die Kennung des Druckauftrags.

Anforderungen an den Standarddrucker senden

Wenn der Druckdienst LP für einen Standarddrucker konfiguriert ist, können Sie Druckaufträge ohne Angabe des Druckernamens senden:

```
$ lp dateiname
```

dateiname steht für den Namen der Datei, die gedruckt werden soll.

Die Datei wird in die Warteschlange des Standarddruckers gestellt, und auf dem Bildschirm erscheint die dem Auftrag zugeordnete *Anforderungskennung*.

Um die Datei `/etc/passwd` auf dem Standarddrucker zu drucken, geben Sie folgendes ein:

```
$ lp /etc/passwd
Anforderungskennung ist buche-8 (1 Datei(en))
$
```

Im Handbuch *User Accounts, Printers, and Mail Administration* ist beschrieben, wie Sie einen Standarddrucker für LP festlegen.

Anforderungen an andere Drucker senden

Wenn kein Standarddrucker zur Verfügung steht oder die Datei auf einem anderen Drucker gedruckt werden soll, geben Sie in der Befehlszeile den Namen des gewünschten Druckers an:

```
$ lp -d druckername dateiname
```

Druckername steht für den Namen eines Druckers, auf den Ihr System zugreifen kann, und *dateiname* für die Datei, die gedruckt werden soll.

Die Datei wird in die Warteschlange des Druckers gestellt, und auf dem Bildschirm erscheint die Anforderungskennung.

Um die Datei /etc/passwd auf dem Drucker ahorn zu drucken, würden Sie also folgendes eingeben:

```
$ lp -d ahorn /etc/passwd
Anforderungskennung ist ahorn-9 (1 Datei(en))
$
```

Wenn Sie versuchen, eine Anforderung an einen Drucker senden, der nicht für Ihr System konfiguriert ist, erscheint eine Fehlermeldung:

```
$ lp -d buche /etc/passwd
UX:lp: Fehler: Ziel "buche" ist dem LP Druck Service
               nicht bekannt.
$
```

Hinweise zur Konfiguration von Druckern finden Sie im Handbuch *User Accounts, Printers, and Mail Administration*. Im Abschnitt „Statusinformationen abrufen" erfahren Sie, wie Sie eine Liste der verfügbaren Drucker abrufen.

Benachrichtigung bei Auftragsabschluß

Wenn Sie eine große Datei drucken lassen und wissen möchten, wann der Ausdruck fertig ist, können Sie LP anweisen, Sie bei Abschluß des Druckauftrags zu benachrichtigen. Folgende Benachrichtigungsarten sind möglich:

- Email-Nachricht
- Meldung im Konsolfenster

Wenn Sie per Email benachrichtigt werden wollen, geben Sie beim Senden der Druckanforderung die Option -m ein:

```
$ lp -m dateiname
```

Soll die Benachrichtigung im Konsolfenster angezeigt werden, verwenden Sie statt dessen die Option -w:

```
$ lp -w dateiname
```

Mehrere Exemplare drucken

Wenn von einer Datei mehrere Exemplare gedruckt werden sollen, geben Sie in der Befehlszeile die Option -n und die Zahl der gewünschten Kopien an. Der Drucker behandelt eine solche Anforderung als einen Auftrag und druckt dementsprechend nur ein Deckblatt.

```
$ lp -nzahl dateiname
```

zahl steht für die gewünschte Anzahl Exemplare und *dateiname* für die Datei, die gedruckt werden soll.

Um vier Exemplare der Datei /etc/passwd auf dem Standarddrucker zu drucken, würden Sie also folgendes eingeben:

```
$ lp -n4 /etc/passwd
Anforderungskennung ist buche-9 (1 Datei(en))
$
```

Befehlsoptionen zu lp

Der Befehl lp unterstützt eine Reihe von Optionen, mit denen Sie den Druckauftrag durch Angabe von Zeichensätzen, Filtern, Titeln, Deckblättern usw. erweitern können. In Tabelle 8-1 sind die am häufigsten verwendeten Optionen zusammengefaßt. Wenn Sie in der Befehlszeile mehrere Optionen angeben, trennen Sie sie durch Leerzeichen und geben vor jeder Option einen Bindestrich (-) ein.

In dem folgenden Befehl wird über Optionen das Druckziel und die Anzahl der zu druckenden Exemplare angegeben und eine elektronische Post bei Auftragsabschluß angefordert:

```
$ lp -d druckername -m -n6 dateiname
```

Tabelle 8-1 Übersicht häufig verwendeter `lp`-Optionen

Option	Beschreibung
-d	Druckziel (Name des zu verwendenden Druckers).
-m	Sendet bei Auftragsabschluß eine Email-Nachricht.
-n	Druckexemplare (Anzahl der zu druckenden Kopien).
-t	Druckt einen Titel auf das Deckblatt.
-o nobanner	Unterdrückt die Ausgabe des Deckblatts.
-h	Druckt auf jede Seite eine Kopfzeile.
-c	Kopiert die Datei vor dem Ausdruck.
-w	Sendet bei Auftragsabschluß eine Meldung an Ihr System.

Eine ausführliche Optionsliste finden Sie im *man Pages(1): User Commands*.

Statusinformationen abrufen

Mit dem Befehl `lpstat` können Sie den Status Ihrer Druckaufträge in der Warteschlange, eine Liste der verfügbaren Drucker und die Anforderungskennungen Ihrer Druckaufträge abrufen.

Status von Druckaufträgen prüfen

Wenn Sie wissen möchten, an welcher Stelle Ihre Druckaufträge in der Warteschlange stehen oder ob sie bereits in Bearbeitung sind, geben Sie folgenden Befehl ein:

```
$ lpstat
```

Daraufhin wird eine Liste aller von Ihnen an den Standarddrucker gesendeten Aufträge angezeigt, die noch nicht abgeschlossen sind.

Im folgenden Beispiel befindet sich in der Warteschlange des Druckers Buche eine Datei, die vom System Eiche gesendet wurde

```
$ lpstat
buche-10           fred          1261   12. März  17:34 auf eiche
$
```

Im ersten Feld wird die Anforderungskennung angezeigt. Danach folgen der Name des Benutzers, der die Anforderung gesendet hat, die Größe der Datei in Byte und Datum und Zeitpunkt der Anforderung.

Verfügbare Drucker

Um festzustellen, auf welche Drucker Sie zugreifen können, geben Sie folgendes ein:

```
$ lpstat -s
```

Daraufhin wird zunächst der Status des Druckspoolers und dann eine Liste der verfügbaren Systeme und Drucker angezeigt.

Im folgenden Beispiel kann auf zwei Netzwerkdrucker zugegriffen werden: buche (gleichzeitig der Standarddrucker) und ahorn:

```
$ lpstat -s
Scheduler läuft
Standardziel für System: buche
System für buche: eiche
System für ahorn: ulme
$
```

Statuszusammenfassung

Mit der Option -t können Sie den Status des gesamten Drucksystems abrufen:

```
$ lpstat -t
```

Daraufhin werden auf dem Bildschirm alle verfügbaren Statusinformationen zu LP angezeigt.

Im folgenden Beispiel befinden sich keine Druckaufträge in der Warteschlange. Angegeben ist auch, seit wann die Drucker zur Verfügung stehen und seit wann sie Druckaufträge entgegennehmen:

```
$ lpstat -t
Scheduler läuft
Standardziel für System: buche
System für ahorn: ulme
buche Anforderungen werden akzeptiert seit Mittwoch 2. Januar
1992 18:20:10 MET
ahorn Anforderungen werden akzeptiert seit Montag 4. März 1992
15:53:47 MET
Drucker buche ist nicht belegt. aktiviert seit Mittwoch 2. Januar
1992 18:20:22 MET. verfügbar.

Drucker ahorn ist nicht belegt. aktiviert seit Montag 4. März
1992 15:53:44 MET. verfügbar.
$
```

Druckerstatus

Um Statusinformationen zu den verfügbaren Druckern anzuzeigen, geben Sie den Befehl lpstat mit der Option -p ein. Damit stellen Sie fest, ob der Drucker aktiv oder im Wartezustand (idle) ist, seit wann er zur Verfügung steht und ob er Druckaufträge entgegennehmen kann.

```
$ lpstat -p
```

Im folgenden Beispiel sind zwei Drucker verfügbar, die sich beide im Wartezustand befinden, da die Warteschlange leer ist. (Stehen Aufträge in der Warteschlange, werden sie ebenfalls angezeigt.)

```
$ lpstat -p
Drucker buche ist nicht belegt. aktiviert seit Mittwoch 2. Januar
1992 18:20:22 MET. verfügbar.
Drucker ahorn ist nicht belegt. aktiviert seit Montag 4. März 1992
15:53:44 MET. verfügbar.
$
```

Um Statusinformationen zu einem bestimmten Drucker anzuzeigen, geben Sie nach der Option `-p` den Namen des gewünschten Druckers ein:

```
$ lpstat -p druckername
```

Druckerkonfiguration

Wenn Sie den Befehl `lpstat` mit der Option `-p` *und* der Option `-l` (long) eingeben, werden die Konfigurationsdaten (einschließlich Druckertyp) der verfügbaren Drucker angezeigt:

```
$ lpstat -p -l
```

Wenn Sie die Konfigurationsdaten eines bestimmten Druckers sehen möchten, geben Sie nach der Option `-p` den Namen des Druckers ein.

Im folgenden Beispiel werden die Konfigurationsdaten des Druckers buche angezeigt:

```
$ lpstat -p buche -l
Drucker buche ist nicht belegt. aktiviert seit Mittwoch 2. Januar
1992  18:20:22 MET. verfügbar.
        Format eingehängt:
        Inhaltstypen: PS
        Druckertypen: PS
        Beschreibung:
        Verbindung: direct
        Benutzer zulässig:
                (alle)
        Formate zulässig:
                (keine)
        Meldungszeile nicht benötigt
        Zeichensätze:
                (keine)
        Standard-Laufweite:
        Standard-Seitengröße:
$
```

Befehlsoptionen zu `lpstat`

Mit dem Befehl `lpstat` können, je nach den verwendeten Befehlsoptionen, sehr unterschiedliche Informationen abgerufen werden. Die am häufigsten benutzten Optionen sind in Tabelle 8-2 zusammengefaßt. Wenn Sie Optionen kombinieren (dabei brauchen Sie keine bestimmte Reihenfolge einzuhalten), trennen Sie sie durch Leerzeichen und stellen jeder Option einen Bindestrich (–) voran.

Um zum Beispiel eine Aufstellung aller Konfigurationsdaten eines bestimmten Druckers anzuzeigen, geben Sie folgendes ein:

```
$ lpstat -p druckername -l
```

Tabelle 8-2 Übersicht häufig verwendeter `lpstat`-Optionen

Option	Beschreibung
-a	(Accept) Zeigt, ob die verfügbaren Drucker Aufträge entgegennehmen.
-c	(Class) Zeigt Druckerklassen und die dazugehörenden Drucker.
-d	(Destination) Gibt den Namen des Standarddruckers aus.
-f	(Forms) Zeigt die verfügbaren Druckformulare.
-o	(Output) Zeigt den Ausgabestatus.
`-p [`*liste*`][-D][-l}`	(Printer / Description / Long list) Gibt den Druckerstatus in Kurz- oder Langform aus.
-r	(Request) Zeigt den Status des Drucker-Schedulers.
-R	Zeigt die Position eines Druckauftrags in der Warteschlange.
-s	(Status) Gibt eine Statusübersicht aus.
-S	(Sets) Zeigt die verfügbaren Zeichensätze.
`-u [`*benutzername*`]`	(User) Zeigt die Druckaufträge eines bestimmten Benutzers.
-v	Zeigt die verfügbaren Geräte.

Eine Liste aller Druckoptionen finden Sie in den *man Pages(1): User Commands*.

Druckanforderungen abbrechen

Mit dem Befehl `cancel` brechen Sie einen Druckauftrag ab, der in der Warteschlange steht oder bereits in Bearbeitung ist. Dazu benötigen Sie die Anforderungskennung, die immer aus dem Namen des Druckers, einem Bindestrich und der Nummer des Druckauftrags besteht. Die Anforderungskennung wird beim Absetzen eines Druckauftrags angezeigt und kann auch mit dem Befehl `lpstat` abgerufen werden.

Ein Druckauftrag kann nur von dem Benutzer, der ihn gestartet hat, oder von einem als `root` oder `lp` angemeldeten Benutzer abgebrochen werden.

Abbruch mit Anforderungskennung

Um einen Druckauftrag abzubrechen, geben Sie folgendes ein:

```
$ cancel anforderung
```

anforderung steht für die Anforderungskennung des Druckauftrags.

Der Auftrag wird abgebrochen, es erscheint eine Statusmeldung, und der Drucker beginnt mit der Bearbeitung des nächsten Auftrags in der Warteschlange.

Im folgenden Beispiel werden zwei Druckaufträge abgebrochen:

```
$ cancel buche-3 buche-4
Anforderung "buche-3" abgebrochen
Anforderung "buche-4" abgebrochen
$
```

Abbruch mit Druckername

Um den gerade bearbeiteten Druckauftrag abzubrechen, genügt es, wenn Sie den Befehl `cancel` zusammen mit dem Namen des Druckers eingeben. (Voraussetzung ist natürlich, daß Sie den aktuellen Auftrag gestartet haben.)

```
$ cancel druckername
```

Der Auftrag wird abgebrochen, es erscheint eine Statusmeldung, und der Drucker holt den nächsten Auftrag aus der Warteschlange.

Zum Beispiel:

```
$ cancel buche
Anforderung "buche-3" abgebrochen
$
```

Wenn der Drucker „hängt“ oder ein anderes Problem vorliegt, kann sich der Systemverwalter als `root` oder `lp` anmelden und den aktuellen Druckauftrag mit `cancel` abbrechen.



Arbeiten im Netzwerk 9 

Als *Netzwerk* bezeichnet man eine Gruppe von Computern, die direkt oder indirekt miteinander verbunden sind und Daten austauschen können. Wenn Ihre Workstation an ein Netzwerk angeschlossen ist, stehen Ihnen auch die Ressourcen anderer Netzwerksysteme zur Verfügung. Um auf ein anderes System zuzugreifen, können Sie sich bei ihm anmelden oder auf Ihrem eigenen System Befehle aufrufen, die dann auf dem entfernten System ausgeführt werden.

In diesem Kapitel werden folgende Themen behandelt:

- Allgemeine Hinweise zu Netzwerken
- Anmelden bei entfernten Systemen
- Kopieren von Dateien auf entfernten Systemen
- Ausführen von Befehlen auf entfernten Systemen
- Abrufen von Statusinformationen zu entfernten Systemen

Wenn Ihr System nicht an ein Netzwerk angeschlossen ist, sind diese Informationen für Sie zwar nicht von unmittelbarem Interesse, es empfiehlt sich jedoch, den Inhalt des Kapitels zumindest zu überfliegen, um sich einen Überblick über die Vorteile vernetzter Umgebungen zu verschaffen.

9

Allgemeine Hinweise zu Netzwerken

In einem Netzwerk tauschen die angeschlossenen Systeme Daten über ein Übertragungsmedium (z. B. ein Kabel oder eine Telefonleitung) aus. Netzwerke werden meist in zwei Kategorien eingeteilt: *Lokale Netzwerke* (`LAN`s) sind räumlich in der Regel auf ein oder mehrere Gebäude begrenzt, während *Weitverkehrsnetze* (`WAN`s) einen Radius von mehreren tausend Kilometern haben können. Mitunter wird auch von *Campus Area Networks* (`CAN`s) gesprochen, die von der Größe her zwischen den beiden anderen Gruppen liegen.

Ein Netzwerk, das seinerseits aus mehreren, miteinander verbundenen Netzwerken besteht, wird als *Internetzwerk* bezeichnet. Lokale Netzwerke sind häufig Teil eines Internetzwerks. Da die Unterschiede zwischen Netzwerken und Internetzwerken für den Benutzer meist nicht erkennbar sind, wird in diesem Handbuch generell der Begriff „Netzwerk" verwendet.

Die Kommunikation zwischen den angeschlossenen Systemen wird durch ein *Netzwerkprotokoll* gesteuert, in dem festgelegt ist, wie und in welchem Format Daten über das Netzwerk übertragen werden. Ein *Internetzwerk-Protokoll* regelt die Kommunikation zwischen Netzwerken.

Anmeldung bei anderen Systemen (`rlogin`)

Mit dem Befehl `rlogin` können Sie sich über das Netzwerk bei einem anderen `UNIX`-System anmelden, das an Ihr Netzwerk angeschlossen ist.

Der Befehl wird in folgendem Format eingegeben:

```
$ rlogin systemname
```

systemname steht für den Hostnamen des entfernten Systems.

Wenn Sie zur Eingabe eines Paßworts aufgefordert werden, geben Sie das Paßwort für das entfernte System ein und drücken „Return". Ist der Name Ihres Systems in der Datei `/etc/hosts.equiv` auf dem entfernten System eingetragen, werden Sie als „vertrauenswürdig" eingestuft und können sich ohne Angabe des Paßworts anmelden.

```
$ rlogin garfield
Password: (Geben Sie das Paßwort ein.)
Last login: Mon Jan 6 09:37:55 from calvin
Sun Microsystems, Inc.   SunOS 5.1      October 1992
(Die folgenden Befehle werden auf dem System garfield ausgeführt.)
$ pwd
/home/calvin
$ logout
Connection closed.
$
```

Anmeldung ohne Home-Verzeichnis

Im obigen Beispiel befindet sich Benutzer `calvin` nach der Anmeldung bei `garfield` in seinem Home-Verzeichnis. Wenn Sie sich bei einem System anmelden, auf dem Sie kein Home-Verzeichnis haben, zeigt `rlogin` eine entsprechende Meldung an, und Sie befinden sich danach im Root-Verzeichnis (`/`) des entfernten Systems:

```
$ rlogin pluto
Password:
No directory! Logging in with home=/
Last login: Fri Jan 3 10:21:59 from calvin
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(Die folgenden Befehle werden auf dem System pluto ausgeführt.)
$ pwd
/
$ logout
Connection closed.
$
```

Anmeldung unter einem anderen Benutzernamen

Mit der Option `-l` können Sie sich unter einem anderen Namen bei einer entfernten Workstation anmelden. Dies ist beispielsweise nötig, wenn Sie unter einem anderen Namen auf einem fremden System arbeiten und sich (mit Ihrem Namen) auf Ihrem eigenen System anmelden möchten. In diesem Fall geben Sie den Befehl `rlogin` in folgendem Format ein:

`rlogin` *systemname* `-l` *benutzername*

Im nächsten Beispiel meldet sich Benutzer `jon` von seinem System `garfield` aus als `donald` auf dem System `pluto` an:

```
$ rlogin pluto -l donald
Password:
Last login: Wed Jan 8 07:12:25 from garfield
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(Die folgenden Befehle werden auf dem System lonesome ausgeführt.)
$ pwd
/home/donald
$ logout
Connection closed.
$
```

Wenn Sie sich unter einem anderen Benutzernamen anmelden, befinden Sie sich danach im Home-Verzeichnis dieses Benutzers.

Anmeldung bei einem unbekannten System

Die Namen der Systeme, auf die Sie über das Netzwerk zugreifen können, sind in einer Datenbank eingetragen, die auf Ihrem System gespeichert ist. Wenn Sie versuchen, sich bei einem System anzumelden, dessen Name in der Host-Datenbank nicht enthalten ist, zeigt `rlogin` folgende Meldung an:

```
$ rlogin unbekannt
unbekannt: unknown host
$
```

Verbindungen abbrechen

Normalerweise beenden Sie eine `rlogin`-Sitzung durch Eingabe von `logout`. Läßt sich die Sitzung aus irgendeinem Grund damit nicht beenden, geben Sie am Anfang einer Zeile eine Tilde und danach einen Punkt ein (`~.`). Damit brechen Sie die Verbindung zu dem entfernten System ab und kehren in die eigene Shell zurück.

Wenn Sie sich nacheinander bei mehreren entfernten Systemen anmelden und dann eine der Verbindungen mit `~.` abbrechen, werden alle dazwischenliegenden Verbindungen ebenfalls abgebrochen.

```
$ rlogin jerry
Password:
Last login: Fri Jan 10 09:14:43 from garfield
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(Der folgende Befehl wird auf dem System jerry ausgeführt.)
$ ~.  (Die Tilde wird unter Umständen nicht angezeigt.)
Connection closed.
$
```

Wenn Sie statt dessen nur die letzte Verbindung abbrechen wollen, geben Sie zwei Tilden und einen Punkt ein (`~~.`):

```
$ rlogin pluto
Password:
Last login: Tue Jan 7 08:12:49 from garfield
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(Der folgende Befehl wird auf dem System pluto ausgeführt.)
$ rlogin jerry
Password:
Last login: Tue Jan 7 10:17:40 from pluto
Sun Microsystems, Inc.    SunOS 5.1      October 1992
(Der folgende Befehl wird auf dem System jerry ausgeführt.)
$ ~~.  (Die Tilden werden unter Umständen nicht angezeigt.)
Connection closed.
$
```

Verbindungen unterbrechen

Um eine `rlogin`-Sitzung zu unterbrechen, geben Sie eine Tilde (~) und danach „Ctrl-Z“ ein. Damit halten Sie den `rlogin`-Prozeß an und kehren zu Ihrem lokalen System zurück.

Um die Verbindung wieder zu aktivieren, geben Sie den Befehl `fg` oder ein Prozentzeichen und die Nummer des angehaltenen Prozesses ein. (Wenn Sie keine Prozeßnummer angeben, wird der zuletzt angehaltene Prozeß aktiviert.)

```
$ rlogin pluto
Password:
Last login: Tue Jan 7 08:12:49 from garfield
Sun Microsystems, Inc.    SunOS 5.1     October 1992
(Der folgende Befehl wird auf dem System pluto ausgeführt.)
~^Z  (Die Zeichen ^Z werden unter Umständen nicht angezeigt.)
Stopped
(Der folgende Befehl wird auf dem System garfield ausgeführt.)
$ pwd
/home/jon
$ %
rlogin pluto

(Der folgende Befehl wird auf dem System pluto ausgeführt.)
$ logout
Connection closed.
$
```

Wenn Sie bei mehreren Systemen angemeldet sind, können Sie, ebenso wie beim Abbrechen von Sitzungen, auch hier durch Eingabe von ~~ und „Ctrl-Z“ nur die letzte Verbindung unterbrechen.

Sitzungsdaten anzeigen (`who am i`)

Wenn Sie bei mehreren Systemen angemeldet sind, wissen Sie vielleicht nicht immer genau, in welcher Sitzung Sie sich gerade befinden. Mit dem Befehl `who am i` können Sie den Namen des aktuellen Systems und den Benutzer-namen, mit dem Sie sich bei diesem System angemeldet haben, anzeigen.

Weitere Informationen zu diesem Befehl finden Sie in den *man Pages(1): User Commands*.

Dateien zwischen Systemen kopieren (rcp)

Mit dem Befehl rcp können Sie Dateien von einem Netzwerksystem auf ein anderes kopieren. Dabei wird anhand der Dateien /etc/hosts.equiv und /etc/passwd geprüft, ob Sie über entsprechende Zugriffsrechte auf dem entfernten System verfügen. Die Befehlssyntax ähnelt der des Befehls cp.

Hinweis – Um Unterverzeichnisse und ihren Inhalt zwischen Netzwerksystemen zu kopieren, verwenden Sie den Befehl rcp -r.

Kopieren von einem entfernten System

Wenn Sie Dateien von einem entfernten System auf Ihr lokales System kopieren wollen, geben Sie den Befehl in folgendem Format ein:

rcp *systemname*:*quelle ziel*

systemname ist der Name des entfernten Systems, *quelle* steht für den (die) Namen der zu kopierenden Datei(en), und *ziel* ist der Pfadname des lokalen Verzeichnisses, in das die Dateien kopiert werden sollen.

Im folgenden Beispiel wird die Datei /home/tom/doc/brief auf dem entfernten System jerry in das Verzeichnis /tmp auf dem lokalen System garfield kopiert:

```
$ rcp jerry:/home/tom/doc/brief /tmp
$
```

rcp unterstützt verschiedene Metazeichen und Abkürzungen. Mit dem folgenden Befehl kopieren Sie beispielsweise alle Dateien mit der Endung .doc aus dem Home-Verzeichnis des Benutzers donald auf dem System pluto in das aktuelle Verzeichnis auf dem lokalen System garfield:

```
$ rcp pluto:~donald/*.doc .
$
```

Kopieren auf ein entferntes System

Um Dateien von Ihrem lokalen System auf ein anderes Netzwerksystem zu kopieren, geben Sie den Befehl `rcp` in folgendem Format ein:

`rcp` *quelle systemname:ziel*

quelle steht für den (die) Namen der zu kopierenden Datei(en), *systemname* ist der Name des entfernten Systems, und *ziel* wird durch den Pfadnamen des entfernten Verzeichnisses ersetzt, in das die Dateien kopiert werden sollen.

Im folgenden Beispiel wird die Datei `austin` im Verzeichnis `~/usa/texas` in das Verzeichnis `~donald/geo` auf dem entfernten System `pluto` kopiert (~ steht dabei für Ihr Home-Verzeichnis und `~donald` für das Home-Verzeichnis des Benutzers `donald`):

```
$ rcp ~/usa/texas/austin pluto:~donald/geo
$
```

Weitere Informationen zu dem Befehl `rcp` und den dazugehörigen Optionen finden Sie in den *man Pages(1): User Commands*.

Befehle auf entfernten Systemen ausführen (`rsh`)

Mit dem Befehl `rsh` können Sie einen einzelnen Befehl auf einem entfernten Netzwerksystem ausführen, ohne sich dafür extra bei diesem System anmelden zu müssen. Dieser Befehl kann sehr viel Zeit sparen, wenn auf dem anderen System nur eine ganz bestimmte Aufgabe ausgeführt werden muß.

`rsh` wird in folgendem Format eingegeben:

`rsh` *systemname befehl*

Im folgenden Beispiel wird der Inhalt des Verzeichnisses `/home/tom/gitarre` auf dem System `jerry` angezeigt:

```
$ rsh jerry ls /home/tom/gitarre
collings        gibson          santacruz
fender          martin          taylor
$
```

Ähnlich wie die Befehle `rlogin` und `rcp` fragt auch `rsh` den Inhalt der Dateien `/etc/hosts.equiv` und `/etc/passwd` auf dem entfernten System ab, um zu prüfen, ob Sie zur Ausführung des Befehls berechtigt sind.

Weitere Informationen zu dem Befehl `rsh` und den dazugehörigen Optionen finden Sie in den *man Pages(1): User Commands.*

Benutzerdaten anzeigen (`rusers`)

Mit dem Befehl `rusers` können Sie feststellen, wer auf den Systemen Ihres Netzwerks angemeldet ist. Ausgegeben werden die Systemnamen und die Login-Namen der Benutzer:

```
$ rusers
jako        franz
venus       chris
uranus      nina
jupiter     birgit
mars        doris     sabine     hansm
mond        hans
saturn      wolfgang
zorg        calvin
$
```

Auf dem System `mars` sind derzeit drei Benutzer angemeldet.

Um Informationen zu einem bestimmten System anzuzeigen, geben Sie `rusers` und den Namen des Systems ein:

```
$ rusers mars
mars        doris     sabine     hansm
$
```

Wenn Sie den Befehl `rusers` mit der Option `-l` eingeben, erscheint eine ausführlichere Liste, in der außer den Benutzernamen auch der Terminaltyp, Datum und Zeitpunkt der Anmeldung, der Zeitraum seit der letzten Eingabe (falls länger als eine Minute) und gegebenenfalls der Name des Systems, von dem aus sich der Benutzer angemeldet hat, angegeben ist:

```
$ rusers -l mars
doris        mars:ttyd8     Feb 10 08:12     5:29
sabine       mars:console   Feb 10 09:16
hansm        mars:ttyp0     Feb 10 11:56       36
$
```

Wenn Sie den Befehl ohne einen Systemnamen eingeben, werden die Daten aller entfernten Benutzer angezeigt.

Weitere Informationen zu dem Befehl `rusers` und den dazugehörigen Optionen finden Sie in den *man Pages(1): User Commands*.

Anpassen der Arbeitsumgebung 10



Unter SunOS haben Sie die Möglichkeit, die Arbeitsumgebung in vielerlei Hinsicht an Ihre Anforderungen anzupassen. Dazu ändern Sie den Wert der *Umgebungsvariablen* in den *Konfigurationsdateien* Ihres Systems. Wenn Sie sich beim System anmelden, wird der Inhalt dieser Dateien gelesen und das System entsprechend konfiguriert. Durch Setzen der Umgebungsvariablen können Sie sich also Ihre ganz persönliche Arbeitsumgebung „maßschneidern".

In diesem Kapitel lernen Sie die am häufigsten verwendeten Umgebungsvariablen kennen und erfahren, wie Sie die Konfigurationsdateien bearbeiten. Außerdem wird beschrieben, wie Sie SunOS-Befehlen Aliasnamen zuweisen, die Shell-Eingabeaufforderung ändern, Standard-Zugriffsrechte für Ihre Dateien festlegen und die OpenWindows-Schriften anpassen.

Konfigurationsdateien

In welchen Dateien Ihre Systemkonfiguration festgelegt ist, hängt davon ab, welche Shell bei der Systeminstallation als Standard-Shell spezifiziert wurde. Unter SunOS wird normalerweise die Bourne-Shell verwendet, Sie können jedoch auch mit der C- oder Korn-Shell arbeiten. Jede dieser Shells hat eigene Konfigurationsdateien.

Wenn Sie nicht sicher sind, mit welcher Standard-Shell (*Login-Shell*) Sie arbeiten, gehen Sie folgendermaßen vor:

1. **Geben Sie** `echo $SHELL` **ein:**

```
$ echo $SHELL
/bin/sh
```

2. **Sehen Sie sich die Ausgabe des Befehls an:**

- `/bin/sh` – Ihre Login-Shell ist die Bourne-Shell.
- `/bin/csh` – Ihre Login-Shell ist die C-Shell.
- `/bin/ksh` – Ihre Login-Shell ist die Korn-Shell.

Wenn Sie sich das erste Mal anmelden, wird – unabhängig von der verwendeten Shell – normalerweise die Systemprofildatei `/etc/profile` ausgeführt. Diese Datei gehört dem Systemverwalter und kann von anderen Benutzern zwar gelesen, aber nicht verändert werden.

Nach Ausführung des Systemprofils greift das System auf das *Benutzerprofil* zu. Das Benutzerprofil besteht aus einer oder mehreren Konfigurationsdateien, in denen Ihre Arbeitsumgebung definiert ist. In der OpenWindows-Umgebung werden diese Dateien beispielsweise jedesmal gelesen, wenn Sie eine neue UNIX- oder Kommando-Shell starten.

Das Benutzerprofil setzt sich, je nach verwendeter Login-Shell, aus folgenden Dateien zusammen:

- `.profile` (Bourne- und Korn-Shell)
- `.login` und `.cshrc` (C-Shell)

Diese Dateien befinden sich alle in Ihrem Home-Verzeichnis.

Umgebungsvariablen

Wenn Sie Ihre Arbeitsumgebung nur für die Dauer der aktuellen Arbeitssitzung ändern wollen, können Sie das direkt über die Befehlszeile tun. Sollen die Änderungen dagegen für einen längeren Zeitraum gelten, speichern Sie die gewünschten Einstellungen mit Hilfe von Umgebungsvariablen in den Dateien `.profile`, `.login` oder `.cshrc` files.

So zeigen Sie die aktuell auf Ihrem System gesetzten Umgebungsvariablen an:

1. **Geben Sie** `env` **ein, und drücken Sie „Return“.**

```
$ env
HISTORY=100
HOME=/home/hobbes/calvin
HZ=100
LANG=de
LOGNAME=calvin
MAIL=/var/mail/calvin
MANSECTS=\1:1m:1c:1f:1s:1b:2:\3:3c:3i:3n:3m:3k:3g:3e:3x11:3xt:3
w:3b:9:4:5:7:8
PATH=/usr/bin
SHELL=/bin/sh
TERM=sun
TZ=MET
```

Hinweis – Sie können mit dem Befehl `env` auch den Namen der Login-Shell abrufen, da diese in der Umgebungsvariablen `SHELL` angegeben ist. In dem Beispiel oben ist `/bin/sh` (Bourne-Shell) die Login-Shell.

Benutzerprofil

In diesem Abschnitt werden einige der am häufigsten verwendeten Umgebungsvariablen beschrieben. Viele dieser Variablen sind wahrscheinlich schon in Ihrer Profildatei (`.profile` bei der Bourne- und Korn-Shell bzw. `.cshrc` bei der C-Shell) enthalten, die sich im Home-Verzeichnis befindet.

Hinweis – Punktdateien werden mit dem Befehl `ls -la` angezeigt.

In der folgenden Liste sind einige der Umgebungsvariablen aufgeführt, die in Ihrem Benutzerprofil vorhanden sein können. Wie Sie diese Variablen setzen, hängt von der verwendeten Shell ab.

- `CDPATH` – Gibt an, in welchen Verzeichnissen gesucht wird, wenn Sie einen Verzeichnisnamen ohne vollständigen Pfad eingeben.
- `HISTORY` – Legt fest, wie viele Befehle von dem Befehl `history` ausgegeben werden (nur C-Shell).

- HOME – Enthält den absoluten Pfadnamen zu Ihrem Home-Verzeichnis. Das System verwendet diese Variable, um in Ihr Home-Verzeichnis zu wechseln, wenn Sie den Befehl cd ohne Argument eingeben.
- LANG – Definiert die lokale Landessprache (Japanisch, Deutsch, Französisch, Schwedisch oder Italienisch).
- LOGNAME – Enthält Ihren Login-Namen. Diese Variable wird vom System automatisch auf den in der Datei passwd gespeicherten Benutzernamen gesetzt. Informationen zu dieser Datei finden Sie im Handbuch *User Accounts, Printers, and Mail Administration*.
- LPDEST – Enthält den Namen des Standarddruckers.
- MAIL – Gibt an, in welchem Verzeichnis neu ankommende Nachrichten gespeichert werden (normalerweise /var/mail/*benutzername*). Einzelheiten dazu finden Sie in Kapitel 7, „Elektronische Post“.
- MANSECTS – Gibt an, in wie viele Gruppen Online-Dokumente eingeteilt werden können.
- PATH – Enthält die Verzeichnisse, die das System in der angegebenen Reihenfolge durchsucht, um einen Befehl zu finden. Befindet sich ein Befehl nicht in einem der Verzeichnisse des Suchpfades, müssen Sie ihn mit vollständigem Pfadnamen eingeben.

  Diese Variable wird vom System bei der Anmeldung automatisch auf den in der Datei .profile (Bourne- oder Korn-Shell) bzw. .cshrc (C-Shell) angegebenen Suchpfad gesetzt.
- PS1 – Definiert das Aussehen der Shell-Eingabeaufforderung. Standard-mäßig wird in der Bourne- und Korn-Shell ein Dollarzeichen und in der C-Shell ein Prozentzeichen angezeigt. Als Eingabeaufforderung für den Superuser erscheint in allen Shells ein Nummernzeichen (#).
- SHELL – Gibt an, welche Shell von vi und anderen Programmen verwendet wird.
- TERMINFO – Enthält einen Pfadnamen zu einem nicht unterstützten Terminal, das in die Datenbank terminfo aufgenommen wurde. Wenn Sie mit einem Standardterminal arbeiten, brauchen Sie diese Variable nicht zu setzen. Weitere Hinweise zu der Datenbank terminfo finden Sie im Handbuch *User Accounts, Printers, and Mail Administration*.

- TERM – Definiert den aktuell verwendeten Terminaltyp. Wenn Sie einen Editor aufrufen, sucht das System nach einer Datei mit dem in dieser Variablen angegebenen Namen, um die Eigenschaften des Terminals zu ermitteln. Dabei wird zunächst (falls vorhanden) in dem Pfad gesucht, auf den TERMINFO verweist, und dann im Standardverzeichnis /usr/share/lib/terminfo. Findet das System in beiden Verzeichnissen keine Definition, stuft es das Terminal als unintellegentes ASCII-Terminal-ein.
- TZ – Definiert die Zeitzone für die Systemuhr.

Umgebungsvariable PATH *setzen*

Die Umgebungsvariable PATH gibt an, in welchen Verzeichnissen des Datei-systems nach den Befehlen gesucht wird, die Sie in die Befehlszeile eingeben. Dieser Suchpfad kann von Ihnen beliebig erweitert werden.

Ist die Variable nicht gesetzt, müssen Sie jeden Befehl mit vollständigem Pfadnamen eingeben. Nehmen wir an, Sie wollen eine Datei kopieren. Ohne Suchpfad müssen Sie den Befehl als /usr/bin/cp eingeben. Ist das Verzeichnis /usr/bin dagegen im Suchpfad enthalten, geben Sie einfach cp ein. Das System sucht daraufhin in allen Verzeichnissen des Suchpfads nach dem Befehl cp und führt ihn aus, sobald es das richtige Verzeichnis gefunden hat. Wenn Sie alle normalerweise verwendeten Befehlsverzeichnisse in den Suchpfad aufnehmen, können Sie also bei der Eingabe von Befehlen viel Zeit sparen.

Bei der Bourne- und Korn-Shell wird die Variable PATH in der Datei .profile gesetzt, die sich in Ihrem Home-Verzeichnis befindet:

```
PATH=.:/usr/bin:/home/bin
```

home steht für den Pfadnahmen Ihres Home-Verzeichnisses.

Bei der C-Shell wird PATH in der Datei .cshrc gesetzt, die sich ebenfalls in Ihrem Home-Verzeichnis befindet:

```
set path=(. /usr/bin home/bin)
```

home steht für den Pfadnamen Ihres Home-Verzeichnisses.

Hinweis – In der C-Shell können Sie den Pfadnamen des Home-Verzeichnisses auch durch eine Tilde (~) angeben.

Mit folgendem Befehl werden die Änderungen der Variablen PATH in der C-Shell aktiviert:

```
example% source .cshrc
```

In der Bourne- oder Korn-Shell geben Sie statt dessen folgendes ein:

```
$ . .profile
```

Aliasnamen (nur C-Shell)

In der C-Shell können Sie häufig verwendeten Befehlen Aliasnamen zuweisen Auf diese Weise können Sie einen Befehl mit Optionen ausführen, ohne die Optionen explizit angeben zu müssen. Der Befehl `rm` entfernt beispielsweise Dateien standardmäßig, ohne vorher eine Bestätigung einzuholen. Dies ist gefährlich, da durch einen einfachen Tippfehler wertvolle Dateien unwiederbringlich verlorengehen können. Wenn Sie den Befehl dagegen mit der Option `-i` eingeben, werden Sie vor dem Löschen zur Bestätigung des Vorgangs aufgefordert. Indem Sie folgende Zeile in die Datei `.cshrc` aufnehmen

```
alias rm  'rm -i'
```

weisen Sie dem Befehl `rm -i` den Aliasnamen `rm` zu, d. h. wenn Sie `rm` eingeben, wird in Wirklichkeit `rm -i` ausgeführt, und das System läßt sich den Löschvorgang bestätigen. (Bei der Eingabe dürfen Sie die Anführungszeichen nicht weglassen, da die Shell den Text nach dem Leerzeichen sonst nicht richtig interpretiert.)

Um die Änderungen an der Datei `.cshrc` sofort wirksam zu machen, geben Sie folgenden Befehl ein:

```
example% source .cshrc
```

Eingabeaufforderung ändern

Mit welchem Befehl Sie die Eingabeaufforderung Ihrer Shell ändern, hängt davon ab, ob Sie mit der Bourne-, Korn- oder C-Shell arbeiten.

Bourne- und Korn-Shell

In der Bourne- und der Korn-Shell ändern Sie die Eingabeaufforderung mit dem Befehl PS1. Hier drei Beispiele:

```
PS1=": "
PS1="‘hostname‘: "
PS1="‘hostname‘{‘id‘}}: "
```

- Mit dem ersten Befehl erzeugen Sie eine Eingabeaufforderung, die aus einem Doppelpunkt und einem Leerzeichen besteht.
- Mit dem zweiten Befehl erzeugen Sie eine Eingabeaufforderung, die aus Ihrem Systemnamen, einem Doppelpunkt und einem Leerzeichen besteht.
- Mit dem dritten Befehl erzeugen Sie eine Eingabeaufforderung, die aus dem Systemnamen, Ihrem Login-Namen in geschweiften Klammern, einem Doppelpunkt und einem Leerzeichen besteht.

Probieren Sie eines der oben gezeigten Beispiele aus. Die neue Einstellung gilt nur während der aktuellen Arbeitssitzung.

Wenn Sie die Eingabeaufforderung auf Dauer ändern möchten, nehmen Sie die Definition in die Datei .profile in Ihrem Home-Verzeichnis auf.

C-Shell

In der C-Shell ändern Sie die Eingabeaufforderung mit dem Befehl set prompt. Hier drei Beispiele:

```
set prompt="% "
set prompt="‘hostname‘\!: "
set prompt="‘hostname‘{‘id‘}}: "
```

- Mit dem ersten Befehl erzeugen Sie eine Eingabeaufforderung, die aus einem Prozentzeichen und einem Leerzeichen besteht.

- Mit dem zweiten Befehl erzeugen Sie eine Eingabeaufforderung, die aus Ihrem Systemnamen und der History-Nummer des letzten Befehls besteht (angezeigt wird also hostname1, hostname2, hostname3 usw.).
- Mit dem dritten Befehl erzeugen Sie eine Eingabeaufforderung, die aus dem Systemnamen, Ihrem Login-Namen in geschweiften Klammern, einem Doppelpunkt und einem Leerzeichen besteht.

Probieren Sie eines der oben gezeigten Beispiele aus. Die neue Einstellung gilt nur während der aktuellen Arbeitssitzung.

Wenn Sie die Eingabeaufforderung auf Dauer ändern möchten, nehmen Sie die Definition in die Datei `.cshrc` in Ihrem Home-Verzeichnis auf.

Weitere Variablen

Es gibt eine Vielzahl weiterer Variablen, die Sie in der Datei `.profile` oder `.cshrc` setzen können. Eine vollständige Liste aller Umgebungsvariablen finden Sie im *man Pages(1): User Commands*. In diesem Abschnitt werden einige der am häufigsten verwendeten Anpassungsmöglichkeiten beschrieben.

Mit `set noclobber` verhindern Sie, daß Dateien beim Kopieren mit `cp` versehentlich überschrieben werden. Diese Variable wird nur von der C-Shell unterstützt. Geben Sie folgendes in die Datei `.cshrc` ein:

```
set noclobber
```

Mit der Variablen `HISTORY` legen Sie fest, wie viele Befehle in der Historienliste (Befehlsprotokoll) gespeichert werden. Der Inhalt der Liste kann mit dem Befehl `history` abgerufen werden, mit dem Sie auch zuvor eingegebene Befehle wiederholen können. Für die C-Shell nehmen Sie folgende Zeile in die Datei`.cshrc` auf (*zahl* ist die Anzahl der zu speichernden Befehle):

```
set history=zahl
```

Für die Bourne- oder Korn-Shell setzen Sie folgende Zeile in die Datei `.profile` (*zahl* ist die Anzahl der zu speichernden Befehle):

```
HISTORY=zahl
```

Standardzugriffsrechte festlegen

Mit dem Befehl `umask` legen Sie Standardzugriffsrechte fest, die automatisch für alle von Ihnen erzeugten Dateien und Verzeichnisse gelten. Wenn die Mitglieder Ihrer Gruppe und die anderen Benutzer des Netzwerks beispiels-weise nur Lese- und Ausführberechtigung (`-rwxr-xr-x`) für Ihre Daten erhalten sollen, können Sie das in den Dateien `.cshrc` und `.profile` bis auf weiteres festschreiben.

`umask` wird, ebenso wie der Befehl `chmod`, zusammen mit einem numerischen Code verwendet, der jedoch nicht absolut zu verstehen ist, sondern die „Differenz" zwischen den aktuellen und den gewünschten Zugriffsrechten angibt.

Wenn Sie `umask` auf 000 setzen, erhalten sämtliche Benutzer (dies schließt Sie ein) Lese- und Schreibberechtigung für neu erstellte Dateien:

`rw-rw-rw-` (`chmod`-Code 666)

Für alle neu erstellten Verzeichnisse gelten folgende Zugriffsrechte (Lesen, Schreiben und Ausführen):

`rwxrwxrwx` (`chmod`-Code 777)

Um den Wert für `umask` zu berechnen, ziehen Sie den `chmod`-Code für die gewünschten Zugriffsrechte) von dem `chmod`-Code für die derzeit geltenden Standardzugriffsrechte ab. Der verbleibende Rest wird mit `umask` eingegeben.

Nehmen wir an, Sie wollen die Standardzugriffsrechte für Ihre Dateien von 666 (`rw-rw-rw-`) auf 644 (`rw-r--r--`) ändern. Subtrahieren Sie 644 von 666. Das Ergebnis (022) ist der numerische Code, den Sie in die Befehlszeile eingeben:

```
umask 022
```

Die drei Ziffern des Codes sind, ebenso wie bei `chmod`, den drei Benutzerkategorien zugeordnet:

- Die erste Ziffer gilt für den Eigentümer.
- Die zweite Ziffer gilt für die Benutzergruppe.
- Die dritte Ziffer gilt für alle weiteren Benutzer.

Tabelle 10-1 zeigt, welche Bedeutung die sieben möglichen Codezahlen haben.

Tabelle 10-1 Berechtigungscodes für `umask`

umask-Code	Zugriffsrechte
0	rwx
1	rw-
2	r-x
3	r--
4	-wx
5	-w-
6	--x
7	--- (keine)

Weitere Informationen zu `umask` finden Sie im *man Pages(1): User Commands.*

OpenWindows-Standardschriften ändern

Die in OpenWindows-Anwendungen angezeigten Textelemente erscheinen in verschiedenen Standardschriften, die Sie bei Bedarf ändern können.

Schriftstil und Schriftgröße

Die Standardschrift für OpenWindows-Fenster ist Lucida Sans 12 Punkt (`medium`); für Fenstertitel wird Lucida Sans Bold verwendet. Falls gewünscht, können Sie für ein einzelnes Fenster, bestimmte Anwendungen oder alle Anwendungen einen anderen Schriftstil und/oder eine andere Schriftgröße wählen.

Proportionale und nichtproportionale Schriften

Alle Schriften werden in zwei Hauptkategorien eingeteilt – *proportionale* und *nichtproportionale* Schriften. Bei nichtproportionalen Schriften sind alle Zeichen gleich breit, während bei Proportionalschriften die Zeichenbreite differiert. Proportionalschriften sind zwar besser lesbar, für die Bildschirmdarstellung werden jedoch aus technischen Gründen häufig auch nichtproportionale Schriften verwendet.

Schrifttyp festlegen

In der Kommando- und der UNIX-Shell wird eine Proportionalschrift als Standardschrift verwendet. Dies kann mitunter zu Problemen bei der Textausrichtung führen. Wenn Sie häufig mit solchen Problemen konfrontiert sind, sollten Sie eine nichtproportionale Schrift wählen. In den folgenden Beispielen werden für Terminalfenster ausschließlich nichtproportionale Schriften, für Fenster und Fenstertitel dagegen Proportionalschriften verwendet.

Schriftattribute für ein einzelnes Fenster ändern

Um eine Anwendung mit anderen Schriftattributen zu starten, geben Sie den Namen der Anwendung, die Option `-fn` sowie Schriftstil und Punktgröße in eine Befehlszeile ein. Wenn Sie den Befehl mit dem Zeichen & abschließen, steht das Fenster, aus dem Sie die Anwendung gestartet haben, danach weiter zur Verfügung.

```
$ anwendung -fn schriftstil -punktgröße &
```

Die folgenden Beispiele verdeutlichen die Verwendung dieses Befehls:

- Mit dem unten gezeigten Befehl öffnen Sie eine neue Kommando-Shell mit der Proportionalschrift Lucida Sans Typewriter Bold. Da die Schriftgröße nicht spezifiziert ist, wird weiterhin die Standardgröße 12 Punkt verwendet.

```
$ cmdtool -fn lucidasans-typewriter-bold &
```

- In diesem Beispiel öffnen Sie eine neue UNIX-Shell mit der Proportionalschrift Lucida Sans Typewriter Bold und erhöhen die Schriftgröße von 12 auf 14 Punkt. Damit ändern Sie gleichzeitig die Fenstergröße.

```
$ shelltool -fn lucidasans-typewriter-bold-14 &
```

- In diesem Beispiel öffnen Sie ein neues `xterm`-Fenster mit der Schrift `terminal-bold` in der Schriftgröße 16 Punkt:

```
$ xterm -fn terminal-bold-16 &
```

- Mit dem folgenden Befehl starten Sie einen neuen Texteditor mit der Schrift Helvetica Bold 14 Punkt:

```
$ textedit -fn helvetica-bold-14 &
```

Die Option `-fn` wird für alle Anwendungen, Schriftstile und Schriftgrößen verwendet. Im Abschnitt „Verfügbare Schriften anzeigen“ erfahren Sie, wie Sie die in der OpenWindows-Umgebung verfügbaren Schriften abrufen. Diese Schriften können Sie sich mit Hilfe von Demoprogrammen, die im Abschnitt „Schriften auswählen“ vorgestellt werden, auf dem Bildschirm ansehen.

Schriftattribute mehrerer Anwendungen ändern

Um Schriftstil und Schriftgröße dauerhaft zu ändern, nehmen Sie diese Anwendungen mit den gewünschten Attributen über das Fenster „Arbeitsbereich Eigenschaften“ in das Menü „Programme“ auf. Wenn Sie zum Beispiel die Schriftgröße in der Anwendung „Texteditor“ erhöhen wollen, könnten Sie folgende Zeile in das Menü „Programme“ aufnehmen:

```
textedit -fn lucidasans-typewriter-14
```

Sie können für eine Anwendung auch unterschiedliche Schriftattribute definieren und dann je nach Bedarf mit den gewünschten Attributen starten. Wenn die Anwendung „Texteditor“ in der Schriftgröße 12, 14 und 18 Punkt zur Verfügung stehen soll, würden Sie also folgende Zeilen eingeben:

```
textedit -fn lucidasans-typewriter-12
textedit -fn lucidasans-typewriter-14
textedit -fn lucidasans-typewriter-18
```

Um den gewünschten Editor zu starten, wählen Sie einfach die entsprechende Option aus dem Menü „Programme“.

Hinweis – Befehlszeilen, die Sie in das Menü „Programme“ eingeben, sollten nicht mit dem Zeichen & abgeschlossen werden.

Verfügbare Schriften anzeigen

Mit dem Befehl `xlsfonts` können Sie eine Liste aller Schriften abrufen, die in der OpenWindows-Umgebung zur Verfügung stehen. Es empfiehlt sich, diesen Befehl in einer Kommando-Shell einzugeben, da die Liste sehr umfangreich ist und in der Kommando-Shell mit einem Rollbalken durchgeblättert werden kann.

Hinweis – In der von `xlsfonts` generierten Liste sind normalerweise mehr als 400 Schriften aufgeführt. Wenn die Liste auf Ihrem Bildschirm auffallend kürzer ist, sollten Sie Ihren Systemverwalter fragen, ob möglicherweise ein Teil der Schriften nicht installiert worden ist.

Jede Schrift hat einen abgekürzten und einen vollständigen Namen. Der vollständige Name für `lucidasans-typewriter` lautet beispielsweise:

```
-b&h-lucida sans typewriter-medium-r-normal-sans-12-120-72-72-m-
0-iso8859-1
```

In der Schriftenliste sind beide Namen angegeben. Der vollständige Name steht an erster Stelle. Wenn in diesem Kapitel von *schriftname* die Rede ist, verwenden Sie den abgekürzten Namen.

Nach Auswahl einer Schrift fahren Sie fort wie im Abschnitt „Schriftstil und Schriftgröße“ beschrieben.

Schriften auswählen

Sie können nach Belieben andere Schriften ausprobieren, die nicht in den Beispielen erwähnt sind. Um die Liste der verfügbaren Schriften anzuzeigen, rufen Sie sie auf und wählen dann die gewünschte Schrift aus.

Liste der verfügbaren Schriften

Hinweis – Die unten verwendete Schreibweise gibt an, in welcher Reihenfolge Sie die Menüs im Arbeitsbereich öffnen, um das gewünschte Programm zu starten.

Als Hilfe bei der Schriftauswahl stehen drei Programme zur Verfügung, mit denen Sie die verfügbaren Schriften und Schriftgrößen auf dem Bildschirm anzeigen können:

- `Fontview`

  `Fontview` wird über das Menü „Arbeitsbereich“ gestartet: Arbeitsbereich ➤ Programme ➤ Demos.

  Um das Fenstermenü zu öffnen, setzen Sie den Zeiger in das Fenster und drücken MENÜ.

- `Text`

  `Text` wird über das Menü „Arbeitsbereich“ gestartet: Arbeitsbereich ➤ Programme ➤ Demos.

  Um das Fenstermenü zu öffnen, setzen Sie den Zeiger in das Fenster und drücken MENÜ.

- `xfontsel`

  `xfontsel` wird durch Eingabe von `xfontsel &` in die Befehlszeile gestartet:

```
$ xfontsel &
```

Aktualisierung auf OpenWindows Version 3.3 oder höhere Versionen



Solaris arbeitet standardmäßig mit der Benutzerumgebung OpenWindows. Wenn Sie derzeit noch mit einer älteren OpenWindows-Version als 3.3 oder mit der SunView-Umgebung arbeiten und auf OpenWindows Version 3.3 oder eine höhere Version umsteigen wollen, finden Sie hier die dafür nötigen Informationen.

Hinweis – Die SunView-Software wird unter OpenWindows Version 3.3 oder einer späteren Version nicht mehr unterstützt. Im Gegensatz zu früheren OpenWindows-Versionen können Sie SunView nach der Aktualisierung auf Version 3.3 oder einer späteren Version nicht mehr zusammen mit OpenWindows verwenden.

SPARC *Übergang von der SunView-Umgebung*

Die folgenden Hinweise richten sich an Benutzer, die bisher mit SunView gearbeitet haben und auf OpenWindows umsteigen wollen.

SPARC *Die Dateien* `.defaults` *und* `.Xdefaults`

Wenn Sie benutzerspezifische Einstellungen aus der SunView-Umgebung in die OpenWindows-Umgebung übernehmen möchten, können Sie die SunView-Konfigurationsdatei `.defaults` in die von OpenWindows verwendete `.Xdefaults`-Datei umwandeln. Dazu wechseln Sie in das Home-Verzeichnis und rufen das Konvertierungsprogramm `convert_to_Xdefaults(1)` auf:

```
$ cd
$ $OPENWINHOME/bin/convert_to_Xdefaults .defaults
```

Dieses Programm legt in Ihrem Home-Verzeichnis eine Datei namens `.Xdefaults` an, die beim Start von OpenWindows für die Anpassung der Arbeitsumgebung verwendet wird.

Übergang von einer älteren OpenWindows Version zu 3.3 oder höher

Die folgenden Hinweise richten sich an Benutzer, die bisher mit einer ältern OpenWindows Version als Version 3.3 gearbeitet haben.

Die Umgebungsvariable `OPENWINHOME`

Wenn Sie aktuell mit einer OpenWindows-Version arbeiten, die älter als Version 3.3 ist, haben Sie wahrscheinlich die Umgebungsvariable `OPENWINHOME` in einer Konfigurationsdatei gesetzt. Dies wird jedoch, ebenso wie das manuelle Setzen der Variablen mit dem Befehl `setenv`, nicht mehr empfohlen.

Bei Eingabe des Befehls `openwin` wird die Umgebungsvariable `OPENWINHOME` jetzt automatisch auf den Pfad `/usr/openwin` gesetzt, so daß weitere Schritte nicht erforderlich sind.

Wenn Sie die Variable `OPENWINHOME` in der Datei `.profile` oder `.cshrc` in Ihrem Home-Verzeichnis gesetzt haben, löschen Sie diese Zeile oder markieren Sie sie als Kommentar, *bevor* Sie OpenWindows Version 3.3 oder eine spätere Version starten.

Dazu gehen Sie folgendermaßen vor:

1. Öffnen Sie die Datei `.profile` **oder** `.cshrc` **mit einem Texteditor (z. B.** `vi`**).**

2. Geben Sie vor der Variablen, wie unten gezeigt, ein Nummernzeichen (#) ein, oder löschen Sie die gesamte Zeile aus der Datei.
Falls Sie die Datei `.profile` bearbeiten, befolgen Sie Schritt `a`; falls Sie die Datei `.cshrc` bearbeiten, befolgen Sie Schritt `b`.

a. In der Datei `.profile` **sollten die Zeile danach so aussehen:**

```
#OPENWINHOME=/usr/openwin
```

b. In der Datei `.cshrc` **sollten die Zeile danach so aussehen:**

```
#setenv OPENWINHOME /usr/openwin
```

3. Speichern Sie die Datei, und verlassen Sie den Texteditor.

Die Datei `.xinitrc`

In Zusammenhang mit den Dateien `.xinitrc` und `$OPENWINHOME/lib/Xinitrc` ist folgendes zu beachten:

1. Unter OpenWindows Version 2 kopiert das Skript `openwin` die Datei `$OPENWINHOME/lib/Xinitrc` automatisch in eine Datei namens `.xinitrc` in Ihrem Home-Verzeichnis. Unter OpenWindows Version 3.3 geschieht das nicht mehr. Diese Änderung hat folgende Konsequenzen:
   a. `openwin` greift standardmäßig auf `$OPENWINHOME/lib/Xinitrc` zu. Befindet sich in Ihrem Home-Verzeichnis eine Datei namens `.xinitrc`, so hat sie Vorrang vor der Standarddatei.
   b. Beim Start sollte immer die mit OpenWindows Version 3.3 gelieferte `$OPENWINHOME/lib/Xinitrc` verwendet werden. (Sie können allerdings Änderungen, die Sie in Ihrer alten `.xinitrc`-Datei vorgenommen haben, in `$OPENWINHOME/lib/Xinitrc` übernehmen. Hinweise dazu finden Sie im nächsten Abschnitt.)
2. Auf Systemen mit mehreren Bildschirmen muß der OpenWindows-Server `olwm` nicht mehrmals gestartet werden.

Verwendung der korrekten Konfigurationsdatei

Wenn Sie derzeit noch mit einer älteren OpenWindows-Version als 3.3 arbeiten, müssen Sie den Inhalt der Konfigurationsdatei `.xinitrc` in Ihrem Home-Verzeichnis prüfen. Diese Datei enthält unter Umständen benutzerspezifische Optionen.

Um den Status der Datei `.xinitrc` zu prüfen, geben Sie folgende Befehle ein:

```
$ cd
$ ls -a .xinitrc
```

Danach gehen Sie, je nach Befehlsausgabe, folgendermaßen vor:

- Ist eine Datei namens `.xinitrc` nicht vorhanden (der Befehl `ls -a` gibt kein Listing aus), brauchen Sie nichts zu tun, da OpenWindows in diesem Fall die Standard-Konfigurationsdatei verwendet.
- Existiert die Datei `.xinitrc` (der Befehl `ls -a` gibt ein Datei-Listing aus), die jedoch nicht bearbeitet wurde oder keine Änderungen enthält, die übernommen werden sollen, führen Sie Schritt 1 im Abschnitt „.xinitrc löschen oder umbenennen" aus.
- Existiert die Datei `.xinitrc` und enthält sie Änderungen, die erhalten bleiben sollen, führen Sie Schritt 2 im Abschnitt „.xinitrc löschen oder umbenennen" aus.

`.xinitrc` *löschen oder umbenennen*

1. **Um die Datei `.xinitrc` aus Ihrem Home-Verzeichnis zu löschen, geben Sie folgenden Befehl ein:**

```
$ rm .xinitrc
```

2. **Um die Änderungen in der Datei `.xinitrc` zu übernehmen, gehen Sie wie folgt vor:**

   a. **Benennen Sie `.xinitrc` in `.xinitrc.save` um:**

```
$ mv .xinitrc .xinitrc.save
```

b. Kopieren Sie `$OPENWINHOME/lib/Xinitrc` **in** `.xinitrc` **in Ihrem Home-Verzeichnis:**

```
$ cp $OPENWINHOME/lib/Xinitrc $HOME/.xinitrc
```

c. Kopieren Sie die gewünschten Zeilen aus `.xinitrc.save` **in** `.xinitrc`**.**



Vorsicht – Die neue `.xinitrc`-Datei darf nur einen Aufruf für `olwm` und keinen Aufruf für `svenv` enthalten. `$OPENWINHOME/lib/openwin-sys` darf nicht gelöscht werden.

Änderungen am Arbeitsbereich

In früheren Versionen von OpenWindows (vor Version 3.3) wurden Änderungen, die Sie im Fenster „Arbeitsbereich Eigenschaften" vornehmen, in der Datei `.Xdefaults` in Ihrem Home-Verzeichnis gespeichert. OpenWindows ab Version 3.3 speichert diese Einstellungen in der Datei `.OWdefaults`, die sich ebenfalls in Ihrem Home-Verzeichnis befindet. Da diese Datei Vorrang vor `.Xdefaults` hat, kann `.Xdefaults` im Home-Verzeichnis verbleiben.

`.Xdefaults` sollte jedoch *nur* noch für zusätzliche Anpassungen verwendet werden, die über das Fenster „Arbeitsbereich Eigenschaften" nicht möglich sind. Durch Bearbeitung der Datei `.Xdefaults` mit einem Texteditor wie `vi` können Sie beispielsweise Anwendungen konfigurieren, die nicht zu den OpenWindows-Standardanwendungen gehören, oder C-Makros in die Konfiguration aufnehmen. Derartige Anpassungen werden durch die Einstellungen im Fenster „Arbeitsbereich Eigenschaften" nicht beeinflußt.

Menü „Arbeitsbereich“ anpassen

In älteren Versionen der OpenWindows-Software wurde der Inhalt des Untermenüs „Programme“ im Menü „Arbeitsbereich“ durch Bearbeitung der Datei `.openwin-menu` geändert. Bei OpenWindows ab Version 3.3 verwenden Sie dafür das Fenster „Arbeitsbereich Eigenschaften“.

Hinweis – Wenn sich in Ihrem Home-Verzeichnis keine Datei namens `.openwin-menu` befindet, brauchen Sie die folgenden Schritte nicht auszuführen.

Ist die Datei `.openwin-menu` vorhanden, müssen Sie einige Vorbereitungen treffen, um das Menü „Programme“ über „Arbeitsbereich Eigenschaften“ konfigurieren zu können.

Prüfen Sie zunächst, ob `.openwin-menu` folgende Zeile enthält:

```
“Programs”MENU $OPENWINHOME/lib/openwin-menu-programs
```

Wenn ja, löschen Sie die Zeile, und ersetzen Sie sie durch folgenden Eintrag:

```
“Programs”INCLUDE openwin-menu-programs
```

Ist die angegebene Zeile in `.openwin-menu` nicht enthalten, fügen Sie einfach die obenstehende Zeile in die Datei ein.

Damit nehmen Sie das Standardmenü „Programme“ in das Menü „Arbeitsbereich“ auf, so daß es mit „Arbeitsbereich Eigenschaften“ angepaßt werden kann.

Sollten im Menü „Arbeitsbereich“ danach Menüoptionen doppelt auftauchen, entfernen Sie die entsprechenden Zeilen aus der Datei`.openwin-menu`.

Ändern der Tastenbelegung



In diesem Anhang erfahren Sie, wie Sie die Belegung des rechten und linken Funktionstastenblocks vertauschen, um als Linkshänder bequemer mit der Tastatur arbeiten zu können, und wie Sie die Taste „Compose“ deaktivieren.

Weitere Informationen über das Ändern der Maustastenbelegung (z. B. Umstellen der rechten und linken Maustaste für Linkshänder) finden Sie im Handbuch Solaris Benutzerhandbuch.

Compose-Taste deaktivieren

---

x86 – Die Funktion der Taste „Compose“ erzeugen Sie auf x86-Tastaturen durch Eingabe der Tastenkombination „Strg-Shift-F1“ bzw. „Ctrl-Shift-F1“.

Wenn Sie die Taste „Compose“ nicht benötigen, können Sie sie deaktivieren, um falsche Eingaben durch versehentliches Drücken der Taste zu vermeiden. Dazu ermitteln Sie zunächst den Tastaturcode für Multi_key:

```
$ xmodmap -pk | grep Multi_key
```

Das System zeigt eine Meldung in folgendem Format an:

nn 0xff20 (Multi_key)

Wichtig an dieser Meldung ist die zweistellige Codezahl am Anfang der Zeile (hier durch *nn* angegeben). Geben Sie die Codezahl zusammen mit den unten gezeigten Informationen in die Datei `.xinitrc` in Ihrem Home-Verzeichnis ein:

```
xmodmap -e 'keycode nn = NoSymbol'
```

Um die Taste „Compose“ wieder zu aktivieren, markieren Sie den Eintrag in der Datei .xinitrc als Kommentar und starten OpenWindows neu.

SPARC *Umbelegung der Funktionstasten für Linkshänder*

Mit dem in diesem Abschnitt beschriebenen Skript können Sie die Belegung der Tasten auf der rechten und linken Seite der Tastatur (d. h. rechts und links vom Hauptblock) auf Typ 4 und Typ 5 Tastaturen ändern.

SPARC – Beachten Sie bitte, daß die folgenden Abschnitte zur „Umbelegung der Funktionstasten für Linkshänder“ nur für SPARC-Systeme gelten.

SPARC *Tastenbelegung über ein Skript ändern*

So erstellen Sie ein Skript für die Umbelegung der Tastenfunktionen:

1. **Erstellen Sie mit einem Texteditor eine Datei namens `links.data`.**
   Sie können diese Datei in einem beliebigen Verzeichnis anlegen, müssen sich aber bei Ausführung von Schritt 4 in diesem Verzeichnis befinden.

2. **Geben Sie den Text in Table B-1 in die Datei ein.**
   Mit einem Ausrufezeichen beginnende Zeilen sind Kommentare, die nicht unbedingt miteingegeben werden müssen.

3. **Speichern Sie die Datei, und verlassen Sie den Texteditor.**

4. **Geben Sie folgenden Befehl ein:** xmodmap links.data

5. **Klicken Sie im Arbeitsbereich eine beliebige Maustaste.**
   Damit aktivieren Sie die neue Tastenbelegung.

Geben Sie folgende Zeilen in das Skript links.data ein:

Table B-1 Das Skript links.data

```
!
! links.data
!
! Mit diesem Skript weisen Sie xmodmap an, die Funktionen der rechten und linken
! Funktionstasten auf einer Typ 4 Tastatur von Sun zu vertauschen. Das Skript wird
! mit folgendem Befehl ausgeführt:
!
! xmodmap links.data
!
! Die folgenden Kommentare beziehen sich auf die danach angegebenen
! keycode-Anweisungen.
!
! Vertausche L2 (Again) mit R1 (Pause)
! Vertausche L3 (Props) mit R6 (KP_Multiply)
! Vertausche L4 (Undo) mit R4 (KP_Equal)
! Vertausche L5 (Front) mit R9 (KP_9)
! Vertausche L6 (Copy) mit R7 (KP_7)
! Vertausche L7 (Open) mit R12 (KP_6)
! Vertausche L8 (Paste) mit R10 (Left)
! Vertausche L9 (Find) mit R15 (KP_3)
! Vertausche L10 (Cut) mit R13 (KP_1)
!
! Ändere R3 (Break) in L1 (Stop)
! Ändere R2 (Print) in R10 (Left)
! Ändere R5 (KP_Divide) in R12 (Right)
!
! Ändere Linefeed in Control-R
!
```

Table B-1 Das Skript links.data

```
keycode 10 = R1          R1       Pause
keycode 28 = L2           L2       SunXK_Again
keycode 32 = R6           R6       KP_Multiply
keycode 54 = L3           L3        SunXK_Props
keycode 33 = R4          R4        KP_Equal
keycode 52 = L4          L4        SunXK_Undo
keycode 56 = R9          R9        KP_9       Prior
keycode 77 = L5          L5        SunXK_Front
keycode 58 = R7          R7        KP_7       Home
keycode 75 = L6          L6        SunXK_Copy
keycode 79 = Right      R12      KP_6
keycode 100 = L7         L7        SunXK_Open
keycode 80 = Left        R10      KP_4
keycode 98 = L8          L8      SunXK_Paste
keycode 102 = R15        R15    KP_3          Next
keycode 121 = L9         L9      SunXK_Find
keycode 104 = R13        R13    KP_1          End
keycode 119 = L10        L10    SunXK_Cut
keycode 30 = L1          L1      SunXK_Stop
keycode 29 = Left        R10    KP_4
keycode 53 = Right      R12    KP_6

keycode 118 = Control_R
add control = Control_R
```

SPARC *Standardbelegung wiederherstellen*

Wenn Sie OpenWindows verlassen und neu starten, wird automatisch wieder die Standard-Tastenbelegung aktiviert. Dieses Verfahren ist aber sehr umständlich, wenn Sie die Standardbelegung nur vorübergehend

wiederherstellen möchten. Es empfiehlt sich daher, ein zweites Skript zu erstellen, mit dem bei Bedarf auf die Standardbelegung umgeschaltet werden kann.

Dazu gehen Sie folgendermaßen vor:

1. **Erstellen Sie mit einem Texteditor eine Datei namens** rechts.data**.**
   Diese Datei muß im gleichen Verzeichnis wie `links.data` gespeichert werden.

2. **Geben Sie den Text in Table B-2 in die Datei ein.**
   Mit einem Ausrufezeichen beginnende Zeilen sind Kommentare, die nicht unbedingt miteingegeben werden müssen.

3. **Speichern Sie die Datei, und verlassen Sie den Texteditor.**

4. **Geben Sie folgenden Befehl ein:**

```
$ xmodmap rechts.data
```

   Dabei müssen Sie sich in dem Verzeichnis befinden, in dem rechts.data gespeichert ist.

Table B-2 Das Skript rechts.data

```
!
! rechts.data
!
! Mit diesem Skript weisen Sie xmodmap an, die Standardbelegung der rechten und
! linken Funktionstasten auf einer Typ 4 Tastatur von Sun wiederherzustellen. Das
! Skript wird mit folgendem Befehl ausgeführt:
!
! xmodmap rechts.data
!
!Standardbelegung der linken Funktionstasten wiederherstellen:
!
keycode 10 = L2     L2     SunXK_Again
keycode 32 = L3     L3     SunXK_Props
```

Table B-2 Das Skript rechts.data

```
keycode 33 = L4      L4     SunXK_Undo
keycode 56 = L5      L5     SunXK_Front
keycode 58 = L6      L6     SunXK_Copy
keycode 79 = L7      L7     SunXK_Open
keycode 80 = L8      L8     SunXK_Paste
keycode 102 = L9     L9     SunXK_Find
keycode 104 = L10    L10    Sun XK_Cut
!
! Standardbelegung der rechten Funktionstasten wiederherstellen:
!
keycode 28 = R1      R1     Pause
keycode 29 = R2      R2     Print
keycode 30 = R3      R3     Scroll_Lock   Break
keycode 52 = R4      R4     KP_Equal
keycode 53 = R5      R5     KP_Divide
keycode 54 = R6      R6     KP_Multiply
keycode 75 = R7      R7     KP_7          Home
keycode 77 = R9      R9     KP_9          Prior
keycode 98 = Left    R10    KP_4
keycode 100 = Right R12     KP_6
keycode 119 = R13    R13    KP_1          End
keycode 121 = R15    R15    KP_3          Next
!
! Standardfunktion der Linefeed-Taste wiederherstellen:
!
remove control = Control_R
5keycode 118 = Linefeed
```

Netzwerkanwendungen



In diesem Anhang erfahren Sie, wie Sie mit Anwendungen arbeiten, die sich auf anderen Netzwerksystemen befinden.

Hinweis – Die meisten Benutzer benötigen die hier enthaltenen Informationen nicht. Wenn Sie wissen möchten, ob auf entfernten Systemen spezielle Anwendungen zur Verfügung stehen, die für Sie interessant sein könnten, wenden Sie sich an Ihren System- oder Netzwerkverwalter.

Die im Arbeitsbereich von OpenWindows angezeigten Anwendungen laufen normalerweise alle auf Ihrem lokalen System ab. Wenn Ihre Workstation an ein Netzwerk angeschlossen ist, können Sie Anwendungen jedoch auch auf entfernten Systemen starten und auf dem lokalen Bildschirm anzeigen lassen. Auf diese Weise sparen Sie CPU-Kapazität und können unter Umständen auf Anwendungen zugreifen, die lokal nicht zur Verfügung stehen.

In den folgenden Abschnitten wird an einem einfachen Beispiel gezeigt, wie Sie eine Anwendung auf einem anderen System starten und auf Ihrem lokalen System anzeigen. Diese Anweisungen sollten jedoch nur als allgemeine Richtschnur verstanden werden, da der genaue Ablauf von der Konfiguration Ihrer Workstation und des Netzwerks abhängt. Im Abschnitt „Datensicherheit im Netzwerk" finden Sie weitere Hinweise, die beim Zugriff auf andere Netzwerksysteme beachtet werden müssen.

Damit Sie die nachfolgenden Schritte ausführen können, müssen folgende Bedingungen erfüllt sein:

- Sie müssen berechtigt sein, auf das entfernte System zuzugreifen.
- Ihr Home-Verzeichnis muß über NFS auf dem entfernten System eingehängt werden können.
- Die Anwendung und die dazugehörigen Funktionsbibliotheken müssen auf dem entfernten System installiert sein.

Wenn Sie Fragen zu diesen Punkten haben, wenden Sie sich an Ihren Systemverwalter.

Zugriff auf entfernte Anwendungen (`rlogin`)

Bevor Sie auf die Anwendungen eines anderen Systems zugreifen können, müssen Sie die aktuellen Werte Ihrer Umgebungsvariablen prüfen:

- Die Variable `HOME` in Ihrer Shell auf dem entfernten System muß auf Ihr Home-Verzeichnis gesetzt sein.
- Die Variable `DISPLAY` in Ihrer Shell auf dem entfernten System muß auf den Namen des lokalen Systems gesetzt sein.
- Wenn die OpenWindows-Bibliotheken nicht im Verzeichnis `/usr/lib` oder `/usr/local` installiert sind, müssen Sie die Umgebungsvariable `LD_LIBRARY_PATH` auf das entsprechende Verzeichnis setzen (`$OPENWINHOME/lib`).

Das folgende Beispiel zeigt, wie Sie mit `rlogin` eine Kommando-Shell auf einem entfernten System starten. Dabei wird davon ausgegangen, daß Ihr Home-Verzeichnis im Verzeichnis `/home/meinverzeichnis` auf dem entfernten System eingehängt und die OpenWindows-Software auf diesem System im Ver-

zeichnis /usr/openwin installiert ist. Ersetzen Sie bei der Eingabe *meinverzeichnis* und *meinsystem* durch den entsprechenden Verzeichnis- bzw. Systemnamen und cmdtool durch den Namen der gewünschten Anwendung.

```
$ rlogin entferntes_system
    .
    .
(Die folgenden Befehle werden auf dem entfernten System ausgeführt.)
    .
    .
$ HOME=/home/meinverzeichnis
$ DISPLAY=meinsystem:0
$ LD_LIBRARY_PATH=/usr/openwin/lib
$ /usr/openwin/bin/cmdtool &
```

Nach Eingabe der letzten Zeile erscheint auf Ihrem Bildschirm eine Kommando-Shell. Obwohl Sie mit dieser Anwendung genauso arbeiten wie mit lokalen Anwendungen, läuft sie tatsächlich auf dem entfernten System ab. In diesem Beispiel bringt Ihnen der Zugriff auf die entfernte Anwendung zwar keine großen Vorteile (die Kommando-Shell ist auf Ihrem System ebenfalls verfügbar und benötigt nicht sehr viel Rechenkapazität), aber Sie können auf diese Weise auch auf Anwendungen zugreifen, die auf Ihrem System nicht vorhanden sind oder nicht ablaufen können.

Datensicherheit im Netzwerk

Dieser Abschnitt enthält Informationen zur Datensicherheit im Netzwerk, die beim Zugriff auf entfernte Anwendungen beachtet werden sollten:

- Benutzer- und hostspezifische Zugriffskontrollverfahren
- MIT-MAGIC-COOKIE-1 und SUN-DES-1 (Zugriffsprotokolle)
- Ändern der Server-Zugriffskontrolle
- Starten von Awendungen auf entfernten Systemen (oder lokal) mit einer anderen Benutzerkennung

Voraussetzungen

Die Standard-Zugriffskontrolle von OpenWindows ab Version 3.3 muß nur dann geändert werden, wenn eine der folgenden Bedingungen erfüllt ist:

- Sie arbeiten mit einer Anwendung, die mit `Xlib`- oder `libcps`-Bibliotheken aus älteren OpenWindows- oder X11-Versionen (älter als Version 2 bzw. `X11R4`) verknüpft sind.
- Sie arbeiten mit einer Anwendung, die statisch mit OpenWindows Version 2 Bibliotheken verknüpft ist, und wollen das Zugriffsprotokoll `SUN-DES-1` verwenden.
- Sie arbeiten mit einer Anwendung auf einem entfernten Server.

Zugriffskontrollverfahren

Das Zugriffskontrollverfahren legt fest, welche *Clients* (Anwendungen) auf den X11 / NeWS-Server zugreifen dürfen. Versucht eine Anwendung, ohne entsprechende Berechtigung eine Verbindung zum Server herzustellen, wird sie mit einer Fehlermeldung beendet.

Man unterscheidet zwischen *benutzerorientierten* und *hostorientierten* Kontrollverfahren. Benutzerspezifische Verfahren regeln den Zugriff auf die Zugangsberechtigungen (Accounts) der Netzwerkbenutzer, während hostspezifische Verfahren die Zugriffsrechte für ein bestimmtes System abfragen. Die benutzerspezifische Zugriffskontrolle kann beim Start von OpenWindows mit der Option `-noauth` abgeschaltet werden. Weitere Hinweise dazu finden Sie im Abschnitt „Zugriffskontrollverfahren ändern“.

Benutzerorientierte Zugriffskontrolle

Bei einem benutzerorientierten Zugriffskontrollsystem können jedem Netzwerkbenutzer individuelle Zugriffsrechte zugewiesen werden. Greift ein Benutzer auf den Server zu, fragt dieser Autorisierungsdaten ab, die von der Client-Anwendung gesendet werden. Stimmen die Daten mit den auf dem Server gespeicherten Daten überein, erhält der Benutzer Zugang zu dem Server-System.

Hostorientierte Zugriffskontrolle

Bei einem hostorientierten Zugriffskontrollsystem werden Zugriffsrechte nicht an Benutzer, sondern an Hostsysteme vergeben. Da alle Benutzer eines berechtigten Systems auf den Server zugreifen können, bietet dieses Verfahren weniger Schutz vor unberechtigten Zugriffen als benutzerorientierte Verfahren.

Die hostorientierte Zugriffskontrolle wird vor allem aus Gründen der Abwärtskompatibilität weiterverwendet. Anwendungen, die mit `Xlib`- oder `libcps`-Bibliotheken verknüpft sind, die aus älteren OpenWindows- oder X11-Versionen stammen (älter als Version 2 bzw. Release 4), erkennen das neue benutzerorientierte Kontrollverfahren nicht. Damit auch diese Anwendungen auf den Server zugreifen können, muß der Benutzer entweder auf das hostorientierte Verfahren umschalten oder die Anwendungen mit aktuelleren Bibliotheksversionen verknüpfen.

Hinweis – Aus Sicherheitsgründen sollte, soweit möglich, die zweite Methode verwendet werden.

Zugriffsprotokolle

OpenWindows Version 3.3 unterstützt die Zugriffsprotokolle `MIT-MAGIC-COOKIE-1` und `SUN-DES-1`. Diese Protokolle arbeiten mit unterschiedlichen Autorisierungsdaten, die aber auf ähnliche Weise abgefragt werden. Auf einem Server kann immer nur ein Protokoll implementiert sein. OpenWindows arbeitet standardmäßig mit `MIT-MAGIC-COOKIE-1`, das ein benutzerorientiertes Kontrollverfahren verwendet.

MIT-MAGIC-COOKIE-1

Das Zugriffsprotokoll `MIT-MAGIC-COOKIE-1` wurde am Massachussetts Institute of Technology entwickelt. Beim Start eines Systems über den Server wird für den Benutzer ein sogenanntes *Magic Cookie* erzeugt und auf dem Server gespeichert. Versucht dieser Benutzer danach, auf den Server zuzugreifen, sendet der Client das Magic Cookie zusammen mit der Verbindungsanforderung. Der Server vergleicht das Magic Cookie mit dem bei ihm gespeicherten Magic Cookie und stellt die Verbindung nur bei einer Übereinstimmung her.

SUN-DES-1

Das von Sun entwickelte Zugriffsprotokoll SUN-DES-1 basiert auf Secure RPC (Remote Procedure Call). Bei diesem Protokoll sendet der Client mit der Verbindungsanforderung den mit DES (Data Encryption Software) verschlüsselten Netzwerknamen des Benutzers. Der Server entschlüsselt den Namen und prüft, ob der Benutzer registriert ist.

SUN-DES-1 bietet einen höheren Zugriffsschutz als MIT-MAGIC-COOKIE-1, da der systemunabhängige Netzwerkname im Gegensatz zum Magic Cookie von anderen Benutzern nicht für den Serverzugriff verwendet werden kann.

Wenn Sie anderen Benutzern Zugang zu Ihrem Server verschaffen wollen, können Sie ihre Netzwerknamen in die Zugriffsliste des Servers aufnehmen. Einzelheiten dazu finden Sie im Abschnitt „Zugriffsberechtigungen vergeben (SUN-DES-1)".

Standardprotokoll ändern

Beim Start von OpenWindows können Sie über Befehlsoptionen angeben, daß statt MIT-MAGIC-COOKIE-1 das Protokoll SUN_DES-1 verwendet oder überhaupt keine benutzerorientierte Zugriffskontrolle stattfinden soll. Um auf SUN-DES-1 umzuschalten, starten Sie OpenWindows mit folgendem Befehl:

```
$ openwin -auth sun-des
```

Wenn kein benutzerorientiertes Kontrollverfahren verwendet werden soll, geben Sie in der Befehlszeile die Option -noauth an:

```
$ openwin -noauth
```



Achtung – Mit der Option -noauth wird der Zugriffsschutz erheblich reduziert. Da nach der Deaktivierung der benutzerorientierten Zugriffskontrolle nur noch die Berechtigung des zugreifenden Systems abgefragt wird, kann jeder Benutzer, der Zugriff auf Ihr System hat, auch auf den Server zugreifen.

Zugriffskontrollverfahren ändern

Wenn Sie die benutzerorientierte Zugriffskontrolle nicht mit der Option `-noauth` deaktiviert haben (siehe „Standardprotokoll ändern"), sind beide Kontrollverfahren aktiv. Der Server fragt zunächst die Berechtigung des Benutzers und dann die des Systems ab. Bei der Abfrage der Benutzerdaten wird standardmäßig `MIT-MAGIC-COOKIE-1` verwendet, während die Systemdaten gegen eine leere Liste geprüft werden, d. h. tatsächlich wirksam ist nur die benutzerorientierte Zugriffskontrolle. Mit der Option `-noauth` weisen Sie den Server an, dieses Kontrollverfahren abzuschalten und die hostorientierte Zugriffskontrolle durch Aufnahme des lokalen Systems in die Hostliste zu initialisieren.

Das vom Server verwendete Zugriffskontrollverfahren kann über drei Programme geändert werden: `xhost`, `xauth` und `newshost`. Diese Programme greifen auf zwei Binärdateien zu, die von dem Zugriffsprotokoll erzeugt werden und verbindungsspezifische Autorisierungsdaten enthalten. Die eine Datei wird nur vom Server verwendet, die andere befindet sich im Home-Verzeichnis des Benutzers:

- `.Xauthority` Client-Datei

Mit den Programmen `xhost` und `newshost` bearbeiten Sie die Hostliste auf dem Server. Sobald Sie einen Hostnamen in die Liste aufnehmen, wird der Zugriffsschutz reduziert, da nun alle Benutzer dieses Systems auf den Server zugreifen können, ohne sich weiter ausweisen zu müssen (siehe „Hostorientierte Zugriffskontrolle").

Mit dem Programm `xauth` greifen Sie auf die Autorisierungsdaten in der Client-Datei `.Xauthority` zu. Sie können diese Daten beispielsweise einem anderen Benutzer zur Verfügung stellen, der sie dann in seine Client-Datei einträgt, und ihm so den Zugriff auf Ihren Server ermöglichen.

Weitere Hinweise zur Verwendung von `xhost` und `xauth` finden Sie im Abschnitt „Zugriffsberechtigungen vergeben (MIT-MAGIC-COOKIE-1)".

Die Datei .Xauthority

Die Datei .Xauthority enthält Einträge in folgendem Format:

```
Verbindungsprotokoll        Zugriffsprotokoll
Autorisierungsdaten
```

Normalerweise ist als Zugriffsprotokoll nur MIT-MAGIC-COOKIE-1 angegeben, und die erste und dritte Spalte enthalten nur Einträge für das lokale System. Auf dem Host *anyhost* könnte der Inhalt der Datei .Xauthority beispielsweise so aussehen:

```
anyhost:0      MIT-MAGIC-COOKIE-1  82744f2c4850b03fce7ae47176e75
localhost:0    MIT-MAGIC-COOKIE-1  82744f2c4850b03fce7ae47176e75
anyhost/unix:0 MIT-MAGIC-COOKIE-1  82744f2c4850b03fce7ae47176e75
```

Beim Start des Clients wird der entsprechende Eintrag aus der Datei .Xauthority gelesen, und die Daten aus den Spalten 2 und 3 werden zusammen mit der Verbindungsanforderung an den Server gesendet. In der Standardkonfiguration liefern xhost und newshost leere Host-Zugriffslisten zurück und melden, daß die Zugriffskontrolle aktiviert wurde.

Wenn Sie statt des Standardprotokolls das Protokoll SUN-DES-1 gewählt haben, ist in der Datei .Xauthority in Spalte 2 SUN-DES-1 und in Spalte 3 der Netzwerkname des Benutzers angegeben. Der Netzwerkname erscheint in folgendem Format:

```
unix.benutzer_id@NISdomain_name
```

Auf dem Host *anyhost* könnte der Inhalt der Datei .Xauthority beispielsweise so aussehen (unix.*15339@EBB.Eng.Sun.COM* ist der systemunabhängige Netzwerkname des Benutzers):

```
anyhost:0       SUN-DES-1         "unix.15339@EBB.Eng.Sun.COM"
localhost:0     SUN-DES-1         "unix.15339@EBB.Eng.Sun.COM"
anyhost/unix:0  SUN-DES-1         "unix.15339@EBB.Eng.Sun.COM"
```

Hinweis – Wenn Sie Ihren Netzwerknamen nicht kennen, wenden Sie sich bitte an Ihren Systemverwalter.

Zugriffsberechtigungen vergeben (`MIT-MAGIC-COOKIE-1`*)*

Wenn Sie das Zugriffsprotokoll `MIT-MAGIC-COOKIE-1` verwenden, können Sie einen Benutzer folgendermaßen für den Zugriff auf Ihren Server berechtigen:

1. **Extrahieren Sie auf dem Server mit** `xauth` **einen Eintrag im Format** *hostname*:`0` **in eine Datei.**
   In diesem Beispiel wird als Hostname *anyhost* und als Dateiname *xauth.info*: verwendet:

```
$ $OPENWINHOME/bin/xauth nextract - anyhost:0 > $HOME/xauth.info
```

Da die Zeile sehr lang ist, wird sie hier auf zwei Zeilen verteilt. Bei der Eingabe sollte sie jedoch in einer einzigen Zeile erscheinen.

2. **Senden Sie die Datei (z. B. mit der Anwendung „Post“ oder dem Programm** `rcp`**) an den Benutzer, der zugriffsberechtigt werden soll.**

Hinweis – Aus Datenschutzgründen sollten Sie die Datei möglichst per elektronische Post versenden. Wenn Sie `rcp` verwenden, sollten Sie die Datei *nicht* in einem Verzeichnis ablegen, auf das von anderen Benutzern leicht zugegriffen werden kann.

---

3. **Der Empfänger der Datei muß den Eintrag in die Datei** `.Xauthority` **in seinem Home-Verzeichnis aufnehmen.**
   Zum Beispiel:

```
$ $OPENWINHOME/bin/xauth nmerge - < xauth.info
```

Hinweis – Da die Autorisierungsdaten verbindungsspezifisch sind, werden sie bei einem Neustart des Servers ungültig.

Zugriffsberechtigungen vergeben (SUN-DES-1)

Wenn Sie das Zugriffsprotokoll SUN-DES-1 verwenden, können Sie einen Benutzer folgendermaßen für den Zugriff auf Ihren Server berechtigen:

1. **Registrieren Sie den Benutzer mit** xhost **auf dem Server.**
   In diesem Beispiel wird Benutzer *somebody* für das System *myhost*: zugriffsberechtigt:

```
$ xhost + somebody@
```

2. **Der neue Benutzer muß den Eintrag mit** xauth **in die Datei** .Xauthority **in seinem Home-Verzeichnis übernehmen.**
   In diesem Beispiel lautet der Netzwerkname des neuen Benutzers unix.*15339@EBB.Eng.Sun.COM*. Bei der Eingabe muß auf das Pipe-Zeichen ein Leerzeichen folgen.

```
$ echo 'add myhost:0 SUN-DES-1 "unix.15339@EBB.Eng.Sun.COM"' |
$OPENWINHOME/bin/xauth
```

Da die Zeile sehr lang ist, wird sie hier auf zwei Zeilen verteilt. Bei der Eingabe sollte sie jedoch in einer einzigen Zeile erscheinen.

Clients auf entfernten Systemen starten

X- und NeWS-Clients rufen den Namen des Servers aus der Umgebungsvariablen DISPLAY ab. Um Kompatibilität zu älteren Versionen zu gewährleisten, rufen NeWS-Clients auch den Wert der Umgebungsvariablen NEWSSERVER ab.

Hinweis – Bei NeWS-Clients ist zu beachten, daß die Variable NEWSSERVER vor DISPLAY abgefragt wird. Verweisen die beiden Variablen auf unterschiedliche Systeme oder entfernte Server, stellt der NeWS-Client die Verbindung zu dem durch NEWSSERVER angegebenen System her. Vergewissern Sie sich daher vorher, daß beide Variablen auf denselben Namen gesetzt sind, oder setzen Sie NEWSSERVER auf einen Null-String.

So starten Sie eine Client-Anwendung auf einem entfernten oder unter einem anderen Namen auf dem lokalen System:

1. **Berechtigen Sie einen anderen Benutzer zum Zugriff auf den Server.**
   Folgen Sie dazu den Anweisungen im Abschnitt „Zugriffsberechtigungen vergeben (MIT-MAGIC-COOKIE-1)“ oder „Zugriffsberechtigungen vergeben (SUN-DES-1)“.

2. **Setzen Sie** `DISPLAY` **auf den Namen des Hosts, auf dem der Server läuft.**
   In diesem Beispiel heißt der Host *remotehost*:

```
$ DISPLAY=remotehost:0
```

3. **Starten Sie das Client-Programm.**

```
$ client_program&
```

Der Client wird auf dem entfernten System angezeigt.



DECnet Internetworking (DNI)



Dieser Abschnitt beschreibt, wie OpenWindows- und DECwindows-Systeme über das Transportprotokoll NSP DECnet auf die Anwendungen der jeweils anderen Umgebung zugreifen können.

SPARC – Beachten Sie, daß das Kapitel „ DECnet Interworking (DNI)“ nur für Systeme mit SPARC-Prozessor gilt. Zudem wird DECnet Internetworking erst ab DNI 7.x unterstützt.

Dieses Konzept läßt sich in beiden Richtungen realisieren:

- Sie starten einen X11-Client auf einer VAX (unter dem BetriebssystemVMS) und zeigen das Client-Fenster auf einem OpenWindows-System an.
- Sie starten einen X11-Client auf einem OpenWindows-System und zeigen das Client-Fenster auf einer VAX an.

In den folgenden Abschnitten werden beide Möglichkeiten beschrieben.

Vorbereitungen

So bereiten Sie die Systemumgebung für DECnet Internetworking vor:

1. **Setzen Sie die Umgebungsvariable `DNI_X_ENABLE`, um eine Verbindung über DNI aufzubauen.**
   Der OpenWindows-Server und die Client-Bibliotheken verwenden eine dynamisch ladbare Version der DNI-Transportbibliothek `libdni`. Damit `libdni` vom Server und den Client-Bibliotheken geladen werden kann, müssen Sie die Umgebungsvariable `DNI_X_ENABLE` auf das Verzeichnis setzen, in dem `libdni.so` installiert ist.

Im folgenden Beispiel wird davon ausgegangen, daß Sie DNI über `pkgadd` in den Standardpfad geladen haben:

```
$ DNI_X_ENABLE=/opt/SUNWconn/dni/lib
```

2. **Starten Sie den OpenWindows-Server.**
   Der OpenWindows-Server arbeitet standardmäßig mit dem Zugriffsprotokoll `MIT-MAGIC-COOKIE`, bei dem nicht die Berechtigung des zugreifenden Systems, sondern die des Benutzers abgefragt wird. Im Standardmodus geben die Befehle `xhost` und `newshost` nur eine leere Liste aus und melden, daß die Zugriffskontrolle aktiviert ist. Sie können diesen Kontrollmodus beim Start von OpenWindows mit der Option `-noauth` abschalten.

```
$ openwin -noauth
```

3. **Bitten Sie den Verwalter des OpenWindows-Systems, das VAX-System mit dem Befehl `xhost` für die Herstellung einer X11-Verbindung zum OpenWindows-Server zu berechtigen.**
   Damit X11-Clients über DNI auf den OpenWindows-Server zugreifen können, müssen die DECnet-Knotenadressen auf DECnet-Knotennamen abgebildet werden. Dies wird in der NCP-Datenbank vorgenommen, die Sie auf dem VAX-System erstellen und initialisieren.

```
$ xhost decvax::
```

Die beiden Doppelpunkte stehen für das DECnet-Transportsystem.

Entfernte Clients auf OpenWindows-Systemen anzeigen

Um einen X11-Client auf einem VMS-System zu starten und auf Ihrem OpenWindows-System anzuzeigen, melden Sie sich mit dem SunLink-Befehl `dnilogin` auf dem VAX-System an. Danach setzen Sie auf dem VAX-System die Umgebungsvariable `DISPLAY` auf den Hostnamen des OpenWindows-Systems und starten den X11-Client durch Eingabe des Namens (hier *x11_client*). Informationen zu VMS finden Sie im *VMS DECwindows User's Guide, Running Applications Across the Network.*

Zum Beispiel:

```
$ dnilogin decvax
    .
    .
    .
$ define DECW$DISPLAY OW_machine::0
$ spawn/nowait run x11_client
```

Entfernte Clients auf VAX-Systemen anzeigen

Um X11-Clients auf einem OpenWindows-System starten und auf einem DECWindows-Server anzeigen zu können, müssen Sie die Umgebungsvariable `DISPLAY` auf den Hostnamen des VAX-Systems setzen.

Bevor Sie X11-Clients von einer VAX starten können, müssen Sie die OpenWindows-Schriften auf dem DECwindows-Server kompilieren und installieren. Diese Schriften sind Bestandteil von MIT X11R4 oder des optionalen Schriftenpakets für OpenWindows Version 3.3

1. **Installieren Sie die optionalen OpenWindows-Schriften auf dem OpenWindows-System.**
2. **Hinweise dazu finden Sie im** *OpenWindows Server Programmer's Guide*.
3. **Kopieren Sie die Quelldateien in ein Verzeichnis auf der VAX.**

```
$ cd $OPENWINHOME/share/src/fonts/misc
$ dnicp *.bdf 'decvax::[vaxdir]'
```

4. **Kompilieren Sie die Cursor-Schriften auf der VAX.**
   Dabei erhalten Sie Dateien wie `olcursor.decw$font;1` `olglyph10.decw$font;1`...

```
$ font olcursor.bdf
$ font olglyph10.bdf
...
```

5. **Kopieren Sie die Schriften in das Verzeichnis** `sysfont`:

```
$ set def sys$sysroot:[sysfont.decw.user_cursor16]
$ copy [vaxdir]olcursor.decw$font;1 *
```

Hinweis – Um die Schriften in das Verzeichnis `sysfont` kopieren zu können, müssen Sie auf der VAX als „system" angemeldet sein.

---

6. **Wiederholen Sie die Schritte 2 bis 4 für alle Cursor-Schriften und für die Lucida-Schriften in** `$OPENWINHOME/share/src/fonts/75dpi` **und** `$OPENWINHOME/share/src/fonts/100dpi`**.**

Hinweis – Die Lucida-Schriften sollten in `sys$sysroot:[sysfont.decw.user_75dpi]` und in `sys$sysroot:[sysfont.decw.user_100dpi]` installiert werden.

Die folgende Liste zeigt, welche Schriften auf dem DECwindows-Server mindestens vorhanden sein müssen, damit dieser auf OpenWindows-Anwendungen zugreifen kann. Wenn die Standardschriften dieser Anwendungen nicht geändert wurden, müssen keine weiteren Schriften installiert werden.

- olcursor.bdf
- olglyph10.bdf
- olglyph12.bdf
- olglyph14.bdf
- olglyph19.bdf
- luBS08.bdf
- luBS10.bdf
- luBS12.bdf

- luBS14.bdf
- luRS08.bdf
- luRS10.bdf
- luFS12.bdf
- lutBs12.bdf
- lutRS10.bdf
- lutRS12.bdf

7. **Starten Sie den DECwindows-Server neu.**

8. **Prüfen Sie, ob alle Schriften vorhanden sind:**

```
$ DISPLAY=decvax::0
$ xlsfonts | grep Sun     (Cursor-Schriften)
$ xlsfonts | grep Lucida
```

9. **Vergewissern Sie sich, daß Sie den OpenWindows-Knoten über das Menü „Security" im DECwindows Session Manager berechtigt haben, Anwendungen auf dem DECwindows-Server anzuzeigen.**

10. **Starten Sie eine X11-Anwendung (z. B. die OpenWindows-Anwendung „Post").**

```
$ DISPLAY=decvax::0
$ mailtool
```

Hinweis – `DNI_X_ENABLE` muß auf das Verzeichnis mit der DNI-Transport-bibliothek `libdni` gesetzt sein (siehe Schritt 1 im Abschnitt „Vorbereitungen").

Wenn die untenstehende oder eine ähnliche Fehlermeldung erscheint, müssen Sie die angegebene Schrift auf dem DECwindows-Server installieren, um die Anwendung dort anzeigen zu können.

> XView Warnung: Gewählte Schrift '-b&h-lucida-medium-r-*-*-*-80-*-*-*-*-*-*' kann nicht geladen werden (Font Package)

Diese Meldung besagt, daß die Schrift `luRS10.bdf` nicht vorhanden ist und noch installiert werden muß.



Weitere Informationen zu OpenWindows-Schriften finden Sie im *OpenWindows Server Programmer's Guide*.

Einrichten von Druckern und Modems



Einrichten eines Modems auf Ihrem System

Ein *Modem* ist ein Gerät, welches Ihnen das Senden und Empfangen von Informationen über eine Telefonleitung ermöglicht. Die grafische Anwendung „Administration Tool" macht das Einrichten eines Modems auf Ihrem System zu einer einfachen und komfortablen Aufgabe. Gehen Sie wie folgt vor:

1. **Schließen Sie das Modem an.**
   Um ein Modem an Ihr System anzuschließen, müssen Sie in der Regel folgende Schritte durchführen.

   - Schließen Sie das Modem physikalisch an Ihr System an, oder installieren Sie es (falls es sich um eine Modem-Karte handelt).
   - Stellen Sie alle erforderlichen Schalter und / oder die Übertragungsrate (Baudrate), die Anschlüsse und sonstigen Einstellungen auf dem Modem entsprechend den Anleitungen ein.

   Die erforderlichen Informationen zur Ausführung dieser Schritte finden Sie in der mit dem Modem gelieferten Dokumentation des Herstellers sowie in der Installations-Dokumentation zu Ihrem System.

2. **Melden Sie sich als root beim System an.**
   Sie müssen sich als *root* beim System anmelden, um das „Administration Tool“ benutzen zu können (eine Ausnahme hiervon bilden lediglich die Mitglieder der speziellen UNIX sysadmin-Gruppe (GID 14). Nur als Benutzer root haben Sie die Berechtigung, Systemdateien zu ändern.

   Geben Sie den Befehl `su` ein, um sich als *root* anzumelden:

```
$ su
Password:        (Geben Sie hier das Root-Paßwort ein)
#
```

   Falls die Benutzer-Zugangsberechtigung für root paßwortgeschützt ist, müssen Sie das Paßwort kennen. Falls kein Paßwortschutz eingerichtet wurde, brauchen Sie nach der Paßwort-Eingabeaufforderung nur „Return“ zu drücken. Falls Sie das Root-Paßwort für Ihr System nicht kennen, wenden Sie sich an Ihren Systemverwalter.

3. **Starten Sie die Anwendung „Administration Tool“.**
   Geben Sie folgenden Befehl ein:

```
# /usr/bin/admintool &
#
```

   Die grafische Anwendung „Administration Tool“ wird nach kurzer Zeit am Bildschirm angezeigt.

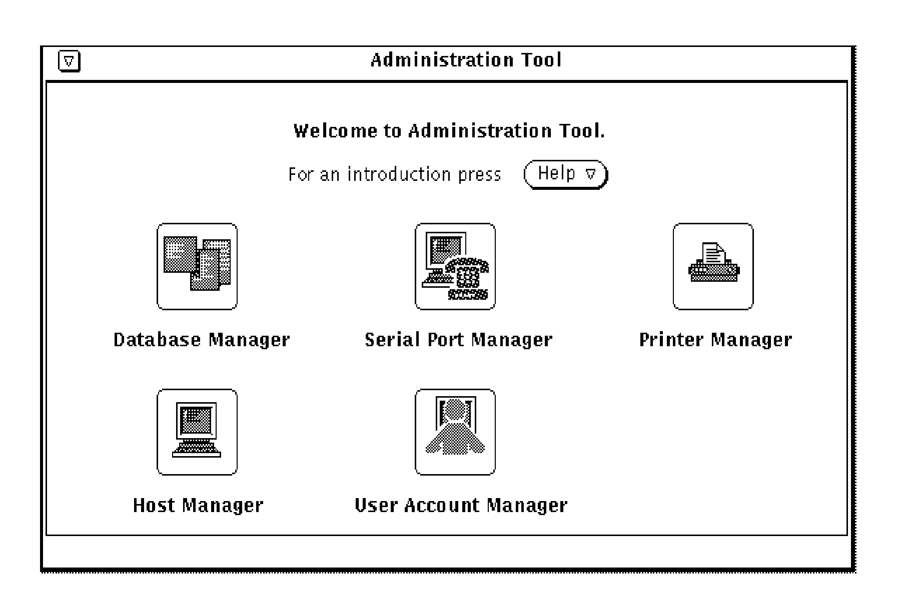


Figure E-1 Die grafische Anwendung „Administration Tool“

4. **Wählen Sie den „Serial Port Manager“.**
   Um den „Serial Port Manager“ zu starten, klicken Sie auf das Symbol des „Serial Port Managers“ im Fenster des „Administration Tools“. Daraufhin wird ein neues Fenster geöffnet, in dem die Steuerelemente und Optionen des „Serial Port Managers“ für die Einrichtung und Verwaltung von Modems und Terminals angezeigt werden.

5. **Befolgen Sie die in der Online-Hilfe angezeigten Anweisungen.**
   Die Online-Hilfe des „Serial Port Managers“ enthält schrittweise Anleitungen zur Benutzung des „Serial Port Managers“ für die Installation eines Modems. Um die Online-Hilfe aufzurufen, wählen Sie „Help...“ im Fenster des „Serial Port Managers“.

Zusätzliche Informationen über Modems

Zusätzliche Informationen über die Installation und Konfiguration von Modems finden Sie im Handbuch *Peripherals Administration*, welches Teil des *Solaris 2.4 System Administrator AnswerBook* ist.

 *E*

Einrichten eines Druckers für Ihr System

Die einfachste Methode, einen Drucker für Ihr System einzurichten, ist die Verwendung der grafischen Anwendung „Administration Tool“. Die folgenden Anleitungen helfen Ihnen beim Einrichten eines lokalen Druckers, der direkt an Ihr System angeschlossen wird. Informationen und Instruktionen für das Einrichten eines Druckers in einem Netzwerk finden Sie im Handbuch *Peripherals Administration*, welches Teil des *Solaris 2.4 System Administrator AnswerBook* ist.

Um einen lokalen, direkt an Ihr System angeschlossenen Drucker einzurichten, gehen Sie wie folgt vor:

1. **Schließen Sie den Drucker an.**
   Hierzu müssen Sie in der Regel folgende Schritte ausführen:

   - Schließen Sie den Drucker physikalisch an das System an.
   - Stellen Sie alle erforderlichen Schalter und/oder die Übertragungsrate (Baudrate), die Anschlüsse und sonstigen Einstellungen auf dem Drucker entsprechend den Anleitungen ein.
   - Schließen Sie den Drucker über sein Netzkabel an eine Netzsteckdose an.

   Das Anschließen des Druckers an Ihr System un das Einschalten der Stromversorgung kann vor oder nach dem Einrichten des Druckers unter Verwendung des „Administration Tools“ erfolgen. Informationen über erforderliche Schaltereinstellungen und Kabel finden Sie in der mit dem Drucker gelieferten Dokumentation des Herstellers sowie im Installationshandbuch zu Ihrem System. In der Regel werden Drucker an eine serielle Schnittstelle angeschlossen. Je nach Anforderung und Konfiguration des Druckers kann in bestimmten Fällen auch die parallele Schnittstelle verwendet werden.

2. **Melden Sie sich als root beim System an.**
   Sie müssen sich als *root* beim System anmelden, um das „Administration Tool“ benutzen zu können (eine Ausnahme hiervon bilden lediglich die Mitglieder der speziellen UNIX sysadmin-Gruppe (GID 14). Nur als Benutzer root haben Sie die Berechtigung, Systemdateien zu ändern.

   Geben Sie den Befehl `su` ein, um sich als *root* anzumelden:

```
$ su
Password:        (Geben Sie hier das Root-Paßwort ein)
#
```

   Falls die Benutzer-Zugangsberechtigung für root paßwortgeschützt ist, müssen Sie das Paßwort kennen. Falls kein Paßwortschutz eingerichtet wurde, brauchen Sie nach der Paßwort-Eingabeaufforderung nur „Return“ zu drücken. Falls Sie das Root-Paßwort für Ihr System nicht kennen, wenden Sie sich an Ihren Systemverwalter.

3. **Starten Sie die Anwendung „Administration Tool“.**
   Geben Sie folgenden Befehl ein:

```
# /usr/bin/admintool &
#
```

   Die grafische Anwendung „Administration Tool“ wird nach kurzer Zeit am Bildschirm angezeigt. (Siehe Figure E-1 on page 191.)

4. **Wählen Sie die Anwendung „Printer Manager“.**
   Um den „Printer Manager“ zu starten, klicken Sie auf das Symbol des „Printer Managers“ im Fenster des „Administration Tools“. Daraufhin wird ein neues Fenster geöffnet, in dem die Steuerelemente und Optionen des „Printer Managers“ für die Einrichtung und Verwaltung von Modems und Terminals angezeigt werden.

5. **Befolgen Sie die in der Online-Hilfe angezeigten Anweisungen.**
   Die Online-Hilfe des „Printer Managers“ enthält schrittweise Anleitungen zur Benutzung des „Printer Managers“ für die Installation eines Druckers. Um die Online-Hilfe aufzurufen, wählen Sie „Help...“ im Fenster des „Printer Managers“.

Zusätzliche Informationen über Drucker

Zusätzliche Informationen über die Installation und Konfiguration von Modems finden Sie im Handbuch *Peripherals Administration*, welches Teil des *Solaris 2.4 System Administrator AnswerBook* ist.

Index

C

D

K

L

M

N

O

P

W

X

Z